US Army Corps
of Engineers
Hydrologic Engineering Center

Determination of a Hydrologic Index for the Russian River Watershed using HEC-ResSim

July 2012

Approved for Public Release. Distribution Unlimited.

PR-85

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
July 2012	Project Report	

4. TITLE AND SUBTITLE	
Determination of a Hydrologic Index for the Russian River Watershed using HEC-ResSim	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER**
Matthew Fleming	**5e. TASK NUMBER**
	5F. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
U.S. Army Corps of Engineers Institute for Water Resources Hydrologic Engineering Center (CEIWR-HEC) 609 Second Street Davis, CA 95616-4687	PR-85
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/ MONITOR'S ACRONYM(S)**
U.S. Army Corps of Engineers San Francisco District 1455 Market St. San Francisco, CA 94103-1398	**11. SPONSOR/ MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT
Distribution authorized to U.S. Government Agencies only.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report was developed by the Hydrologic Engineering Center (CEIWR-HEC) in response to a request from the U.S. Army Corps of Engineers, San Francisco District (CESPN) for assisting in the development of a hydrologic index for the Russian River watershed. The hydrologic index is used to set minimum flows in the Russian River (for biological requirements), which are often met using water released from Lake Mendocino and Lake Sonoma. Currently, the hydrologic index is determined using streamflow measurements in the Eel River watershed, which is north of the Russian River Watershed. The development of new hydrologic index alternatives focused on hydrologic conditions specifically in the Russian River Watershed, like inflow and storage in Lake Mendocino. The hydrologic index alternatives were modeled with an HEC-ResSim model of the Russian River watershed that incorporated operation of both Lake Mendocino and Lake Sonoma.

15. SUBJECT TERMS
Hydrology, HEC-ResSim, watershed, river, hydrologic index, Russian River, flows, Lake Mendocino, Lake Sonoma, streamflow, Eel River, Sonoma County Water Agency, Reservoir System Simulation, model, reservoir operation, channel routing, water supply, historic flows, state variable scripts, cumulative inflow, period-of-record, diversions, distributed losses, water balance

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			
U	U	U	UU	122	**19b. TELEPHONE NUMBER**

Determination of a Hydrologic Index for the Russian River Watershed using HEC-ResSim

July 2012

Prepared for:
US Army Corps of Engineers
San Francisco District
1455 Market St.
San Francisco, CA 94103-1398

Prepared by:
US Army Corps of Engineers
Institute for Water Resources
Hydrologic Engineering Center
609 Second Street
Davis, CA 95616

(530) 756-1104
(530) 756-8250 FAX
www.hec.usace.army.mil

PR-85

Executive Summary

The purpose of this study was to develop potential new alternatives for the hydrologic index in the Russian River watershed. The hydrologic index is used to set minimum flow requirements in the Russian River (for biological requirements). These requirements are often met using water released from Lake Mendocino and Lake Sonoma. Currently, the hydrologic index is determined using the hydrology in the Eel River watershed, which is north of the Russian River watershed. The Sonoma County Water Agency (SCWA) believes that a hydrologic index based on hydrology in the Eel River watershed is not always reflective of the hydrologic index in the Russian River watershed; therefore, new hydrologic index alternatives focusing on hydrologic conditions specifically in the Russian River watershed were evaluated.

This study began with a detailed analysis of the existing HEC-ResSim (Hydrologic Engineering Center's (HEC) Reservoir System Simulation) model of the Russian River watershed. The HEC-ResSim model incorporates reservoir operation, channel routing, and water supply to compute water storage and releases to meet downstream flow requirements (minimum and maximum). After the HEC-ResSim model was "calibrated" to historic flows, model simulations were developed within HRC-ResSim to model the different hydrologic index alternatives. The modeled alternatives were designed to compute the hydrologic index by evaluating a number of conditions in the model, such as cumulative inflow and storage in Lake Mendocino. The hydrologic index was used by the Russian River ResSim model to set minimum stream flows. The hydrologic index alternatives were evaluated by simulating a 98-year period-of-record (1910 – 2008). Boundary condition data for the HEC-ResSim model were provided by SCWA. Unimpaired hydrographs throughout the Russian River Watershed were modeled by the U.S. Geological Survey (USGS) using their Basin Characterization Model (BCM). This hydrologic model used historical weather, climate, and hydrologic data to generate unimpaired flows. Diversions from the Eel River into the Russian River were computed separately by SCWA. SCWA developed estimates for the distributed losses throughout the Russian River Watershed. These losses include SCWA's diversions and all other depletions from the watershed including evapotranspiration by riparian vegetation, aquifer recharge, agricultural diversions, and non-SCWA municipal and industrial (M&I) diversions.

Several hydrologic index alternatives were evaluated using the HEC-ResSim model. These alternatives used cumulative inflow, storage in Lake Mendocino, a water balance in the Upper Russian River, and empirical relationships. A few of the hydrologic index alternatives were developed with the ability to incorporate forecasted inflows from the National Weather Service (NWS). The use of forecasted inflows to make reservoir release decisions was not applied for flood operations, but for setting minimum instream flows. The frequency at which the hydrologic index was computed (weekly, bi-monthly, and monthly) was evaluated in addition to different set of minimum flow schedules (three-schedule and five-schedule). These hydrologic index alternatives, including the HEC-ResSim model, were provided to SCWA for their further development of a preferred hydrologic index.

Table of Contents

Table of Contents

List of Figures

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Project Purpose

A new hydrologic index for the Russian River watershed was developed by the Hydrologic Engineering Center (CEIWR-HEC) in response to a request by the San Francisco District (CESPN), U.S. Army Corps of Engineers (USACE) in coordination with the Sonoma County Water Agency (SCWA). The hydrologic index characterizes the hydrologic regime within a watershed, e.g., Normal, Dry, or Critical. The hydrologic index is used to set minimum instream flow requirements in the Russian River and Dry Creek.

1.2 Scope of this Study

This study was performed to analyze different hydrologic index options for possible use by SCWA. The hydrologic index alternatives were designed to incorporate and evaluate different hydrologic variables within the Russian River watershed in order to determine the hydrologic index. The hydrologic variables that were incorporated into the alternative hydrologic index options include Lake Mendocino cumulative inflow, Lake Mendocino storage, Lake Sonoma storage, losses (riparian, evaporation, and water demand), future/predicted available water, an ensemble of forecasted inflow into Lake Mendocino, and flow in the West Fork Russian River. Hydrologic index options were presented for review to refine and narrow the index options to those preferred by the SCWA. The hydrologic index options were scripted and evaluated in a HEC-ResSim (Reservoir System Simulation) model of the Russian River watershed. The model was provided to SCWA for their continued development of a preferred hydrologic index option.

1.3 Acknowledgements

This study was conducted by Mr. Matt Fleming and Dr. Beth Faber of CEIWR-HEC, USACE, Institute for Water Resources. Chris Delaney of the Sonoma County Water Agency (SCWA) provided data and guidance during the course of this study. Mr. Jeff Harris was the Chief of the Hydrology and Hydraulics Technology Division and Mr. Christopher Dunn was the Director of CEIWR-HEC, during the completion of this study.

Chapter 2
Study Background

2.1 Location, Size, and History

The Russian River watershed, drainage area 1,485 square miles, is located in Northern California within Mendocino and Sonoma Counties. Elevations within the watershed range from zero (0) feet at the ocean outlet to approximately 2,500 feet at the headwaters and 4,500 feet along the eastern edge of the watershed. Higher elevations within the watershed are characterized by forested areas with some concentrated areas of rangeland. The lower portions of the watershed along the Russian River are characterized by agricultural lands with urbanization at concentrated population centers. Major cities along the Russian River include Ukiah, Hopland, Cloverdale, Healdsburg, and Guerneville, as shown in Figure 1. Channel capacities of the Russian River vary from about 7,000 cfs near Ukiah to 35,000 cfs at Guerneville. The Russian River Valley is flanked by two coastal ranges. The Mendocino Range borders the basin to the west and the Mayacamas Mountains to the east. Lake Mendocino and Lake Sonoma are two multipurpose reservoirs in the Russian River watershed that are critical in providing flood control and supplying water to meet water usage requirements and minimum stream flows (for biological needs).

Coyote Valley Dam (Lake Mendocino) is located on the East Fork of the Russian River, about two miles North East of Ukiah, California (Figure 1). Below Coyote Valley Dam, the East Fork of the Russian River flows a very short distance (approximately one mile) until it merges with the West Fork of the Russian River. From the confluence of the East and West Forks, the Russian River travels approximately 100 miles south to southwest until it terminates at the Pacific Ocean. The total drainage area for the watershed above Coyote Valley Dam is approximately 105 square miles. Authorized by the 1950 Flood Control Act, Coyote Valley Dam construction began in July 1956 and was completed January 1959 as part of the Lake Mendocino Project. Lake Mendocino is a multipurpose reservoir utilized for flood control, water supply, hydropower, and recreation. Up to 110,000 acre-feet of water may be carried over into summer in years where spring storms plus water transferred from the Eel River watershed fill the reservoir. Coyote Valley Dam is operated and maintained by USACE. USACE is responsible for operating the dam for flood control purposes. SCWA is responsible for determining reservoir releases for downstream water conservation needs.

Warm Springs Dam (Lake Sonoma) is located on Dry Creek, fourteen miles above the confluence with the Russian River (Figure 1). From the confluence, the Russian River travels approximately thirty-one miles south west until it terminates at the Pacific Ocean. The total drainage area for the watershed above Warm Springs Dam is approximately 130 square miles. Authorized by the 1962 Flood Control Act, Warm Springs Dam construction and the Lake Sonoma Project began in August 1967 and was completed June 1983 (the dam was completed in October 1982 while downstream channel improvements were completed in 1983). Warm Springs Dam was constructed by USACE. The storage space for water conservation is owned by SCWA. Up to 242,000 acre-feet of water may be carried over into summer in years where spring storms fill the reservoir. It is estimated that Lake Sonoma provides 94,500 acre-feet for water supply and minimum biological flows. Warm Springs Dam is operated and maintained by USACE. USACE is responsible for operating the dam for flood control purposes. SCWA is responsible for determining reservoir releases for downstream water conservation needs.

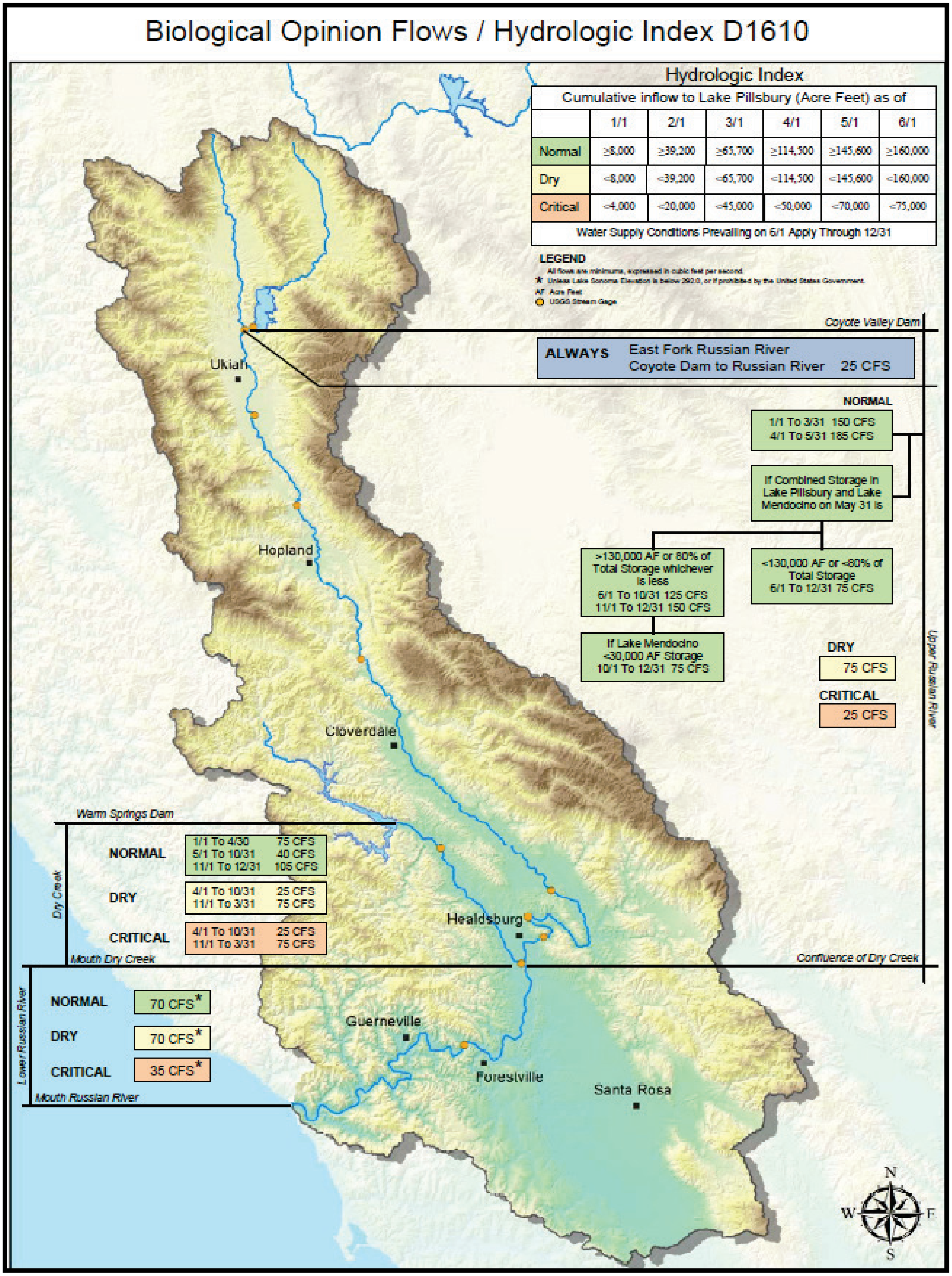

Figure 1. Biological Opinion Minimum Instream Flow Requirements and Current Hydrologic Index in the Russian River Watershed.

The Russian River receives trans-basin diversions from the Eel River through the Potter Valley Project PVP), a hydroelectric facility which is owned and operated by Pacific Gas and Electric (PG&E). PVP diverts water from the Eel River just upstream of Cape Horn Dam into a tunnel which discharges into the East Fork of the Russian River approximately twelve miles upstream of Lake Mendocino. In 2004 PG&E amended its Federal Energy Regulatory Commission (FERC) license for the operation of PVP, which was fully implemented in fall 2006. Prior to the 2006 implementation of the license amendment, an average annual diversion into the Russian River from the Eel River was approximately 288,000 acre-feet per year. Since the license amendment, annual diversions are approximately 165,000 acre-feet per year, which is a forty-two percent reduction in average annual diversions from historic levels.

2.2 Current Hydrologic Index

A hydrologic index can be used to characterize the hydrologic condition within a watershed (e.g., Normal, Dry, or Critical). In the case of the Russian River watershed, a hydrologic index is currently used to determine the minimum instream flow requirements of the Russian River downstream of Coyote Valley Dam and Dry Creek downstream of Warm Springs Dam. These minimum instream flow requirements are set by the California State Water Resources Control Board under Decision 1610 to meet the biological needs and the needs of other beneficial uses within the Russian River. SCWA currently operates under a temporary and interim set of minimum instream flow requirements required under a 2008 Biological Opinion issued by the National Marine Fisheries Service as shown in Table 1. SCWA is currently in the process of completing an environmental study to pursue permanent changes to the minimum flow requirements of Decision 1610. The study will look at a number of different flow alternatives and select a preferred alternative to best meet the needs of all beneficial uses in the watershed.

Table 1. Minimum Flows as Determined by the Hydrologic Index.

Hydrologic Index	**Upper Reach (from Coyote Valley Dam to Healdsburg)**	**Warm Springs Dam to the confluence of Dry Creek and the Russian River**	**Lower Reach (Confluence of Dry Creek and Russian River to the Ocean)**
Normal	150 cfs Jan 1 to Mar 31 185 cfs Apr 1 to May 31 125 cfs Jun 1 to Oct 31 150 cfs Nov 1 to Dec 31	75 cfs Jan 1 to Apr 30 40 cfs May 1 to Oct 31 105 cfs 11/1 to 12/31	70 cfs
Dry	75 cfs	75 cfs Jan 1 to Mar 31 25 cfs Apr 1 to Oct 31 75 cfs Nov 1 to Dec 31	70 cfs
Critical	25 cfs	75 cfs Jan 1 to Mar 31 25 cfs Apr 1 to Oct 31 75 cfs Nov 1 to Dec 31	35 cfs

The current hydrologic index, shown in Figure 1, is based on hydrology of the Eel River. Currently, cumulative inflows into Lake Pillsbury beginning on October 1 and calculated at monthly intervals, from January 1st through June 1st, are used to determine the hydrologic index and set the applicable minimum flow requirements for the Russian River. The index also allows for adjustments to the minimum instream flow requirements of the Upper Russian River (from Coyote Valley Dam to the confluence of Dry Creek) from the hydrologic index. These adjustments to the hydrologic index are known as Dry Spring conditions and are determined based on total storage of Lake Pillsbury and Lake Mendocino on May 31st.

For example, if the cumulative inflow into Lake Pillsbury was 145,600 acre-feet on May 1, then a "Normal" hydrologic index would be in effect. In this case, the minimum flow requirements would be 185 cfs from Coyote Valley Dam to the Confluence of Dry Creek and the Russian River, 40 cfs on Dry Creek, and 70 cfs from the confluence of Dry Creek and the Russian River to the ocean. Furthermore, if that same year had a May 31st combined storage of Lake Pillsbury and Lake Mendocino less than 130,000 acre-feet then minimum flows on the Upper Russian River would be reduced to 75 cfs. If the June 1 cumulative inflow into Lake Pillsbury is greater than 160,000 acre-feet then the rest of the system would stay in a "Normal" hydrologic index with a minimum flow of 75 cfs on Dry Creek and 70 cfs from the confluence of Dry Creek and the Russian River to the ocean.

2.3 Need for a New Hydrologic Index

The watershed above Lake Mendocino contributes approximately 110,000 acre-feet per year from rainfall-runoff. Diversions from the Eel River through PVP make up a large percentage of total annual inflow into Lake Mendocino. With the large reductions in annual diversions due to the recent license amendment discussed above, a hydrologic index based on inflows into Lake Pillsbury no longer reflects the hydrologic condition in the Russian River watershed. There is a need to develop a new hydrologic index that appropriately considers current conditions within the Russian River watershed.

Several new hydrologic index options were evaluated as part of this study (Table 2 contains a list with descriptions). The first eight hydrologic index options were initially developed

Table 2. List of Hydrologic Index Options Evaluated.

	Hydrologic Index	Description
1	Lake Mendocino Cumulative Inflow (monthly evaluation)	Cumulative Inflow into Lake Mendocino is evaluated on the 1st of the month from January 1 through June 1. Storage on May 31 in Lake Mendocino is evaluated.
2	Lake Mendocino Storage (monthly evaluation)	Storage in Lake Mendocino is evaluated on the 1st of the month.
3	Lake Mendocino Cumulative Inflow (weekly evaluation)	Cumulative Inflow into Lake Mendocino is evaluated on a weekly basis from January 1 through June 1. Storage on May 31 in Lake Mendocino is evaluated.
4	Lake Mendocino Storage (weekly evaluation)	Storage in Lake Mendocino is evaluated on a weekly basis.
5	Separate Index for Lake Mendocino (inflow) and Lake Sonoma (storage)	Separate hydrologic index options were developed at both Lake Mendocino and Lake Sonoma. The hydrologic index determining for Lake Mendocino was used to set minimum flows in the upper river and the hydrologic index determined for Lake Sonoma was used to set minimum flows in the lower river (downstream of Healdsburg).
6	Water Balance	A water balance using losses, minimum flows, and projected inflows is evaluated to determine the hydrologic index.
7	Cumulative Inflow + Water Balance	Cumulative inflow into Lake Mendocino is evaluated from January through May and the water balance approach is used for the remainder of the water year.
8	Water Balance using an Ensemble of Forecasted Inflows	The water balance index was modified so that the all 98 years in the period were used instead of project inflows.
9	Lake Mendocino Storage (Bi-Monthly Evaluation)	Storage in Lake Mendocino is evaluated on the 1st and 16th of the month.
10	Future Available Water	An empirical relationship was developed by SCWA relating the hydrologic index to storage in Lake Mendocino and future (predicted) inflow.
11	Modified Storage	Storage in Lake Mendocino is evaluated along with the two-week cumulative runoff in the West Fork Russian River and the predicted inflow from PVP.

and presented to SCWA and the technical advisory group (made up of engineers from the U.S. Geological Survey (USGS), National Weather Service (NWS), and California State Water Resources). After input from SCWA, the list of preferred hydrologic index options was reduced to three: a Lake Mendocino Storage index evaluated semi-weekly, an index computed using an empirical relationship to estimate the amount of available water, and a modified storage index. The empirical and modified storage index alternatives were developed by SCWA with support from CEIWR-HEC. HEC-ResSim was used to evaluate each hydrologic index alternative. The HEC-ResSim model was used to evaluate the effects that the different hydrologic index alternatives had on meeting minimum stream flows. The model was used to simulate a 98-year period-of-record, (1910 – 2008), of stream flows and reservoir operation in the Russian River watershed.

Chapter 3
HEC-ResSim Model of the Russian River Watershed

3.1 HEC-ResSim Model Overview

The Russian River HEC-ResSim model was developed by SCWA to be used as a planning tool to simulate the effects of various climatic conditions, levels of demand, and operational criteria on the water supply available for use by SCWA and others. The HEC-ResSim model calculates the releases that must be made from Lake Mendocino and Lake Sonoma, taking into account USACE flood control operations criteria, Decision 1610 minimum streamflow requirements, and the flow requirements of the Biological Opinion. Output from the HEC-ResSim model includes flows at discrete locations (or "junctions") within the Russian River watershed.

The model incorporates 98 complete years of hydrologic data (1910 - 2008), represented as daily unimpaired tributary flows into the Russian River and Dry Creek. Unimpaired flows are the "natural" flows, unaffected by man-made influences, such as water diversions or reservoir operations. These unimpaired flows, which form the basis of the hydrology in the model, were developed by USGS using their Basin Characterization Model (BCM). This hydrologic model uses historical weather, climate, and hydrologic data to generate the unimpaired flows that are passed to the HEC-ResSim model.

Diversions from the Eel River into the Russian River are defined explicitly in the HEC-ResSim model. These diversions are computed separately using the Eel River Model Version 2.5[1]. These diversions are not based on historical diversions from 1910 to 2008. In the Fall of 2006, operations of the PVP changed based on PG&E's interpretation of its FERC license; therefore, historical diversions would not be representative of current operations. To determine current average operations of the PVP, PG&E diversions from the Eel River were analyzed from October 1, 2006 to January 31, 2011. Using the Eel River Model and the results of the operations analysis, input datasets were developed for the HEC-ResSim model to approximate current PVP operating conditions.

Another major component of the HEC-ResSim model is the distributed losses throughout the Russian River system. These losses include not only the Water Agency's diversions, but all other depletions from the watershed including evapotranspiration by riparian vegetation, aquifer recharge, agricultural diversions, and non-Water Agency municipal and industrial diversions. Much like the unimpaired flow datasets, system losses were aggregated between each junction. System losses not associated with the SCWA's diversions were estimated through an analysis of historical municipal and industrial data, flow gage data, and climate data. Because the model calculates the reservoir releases necessary to meet minimum stream flow requirements, all water uses in the watershed are satisfied by simulated flow releases, not just demands of the SCWA's transmission system.

Figure 2 shows a schematic of the Russian River HEC-ResSim model, and Table 3 contains a list of junctions. The HEC-ResSim model divides the Russian River and Dry Creek into

[1] *This model was developed by Natural Resources Consulting Engineers, Inc. on behalf of the U.S. Department of the Interior for the FERC license amendment of the PVP in 2004. The model was further refined in 2008 by SCWA in collaboration with the Round Valley Indian Tribes to account for diversion restrictions through the PVP as a result of the 2004 license amendment.*

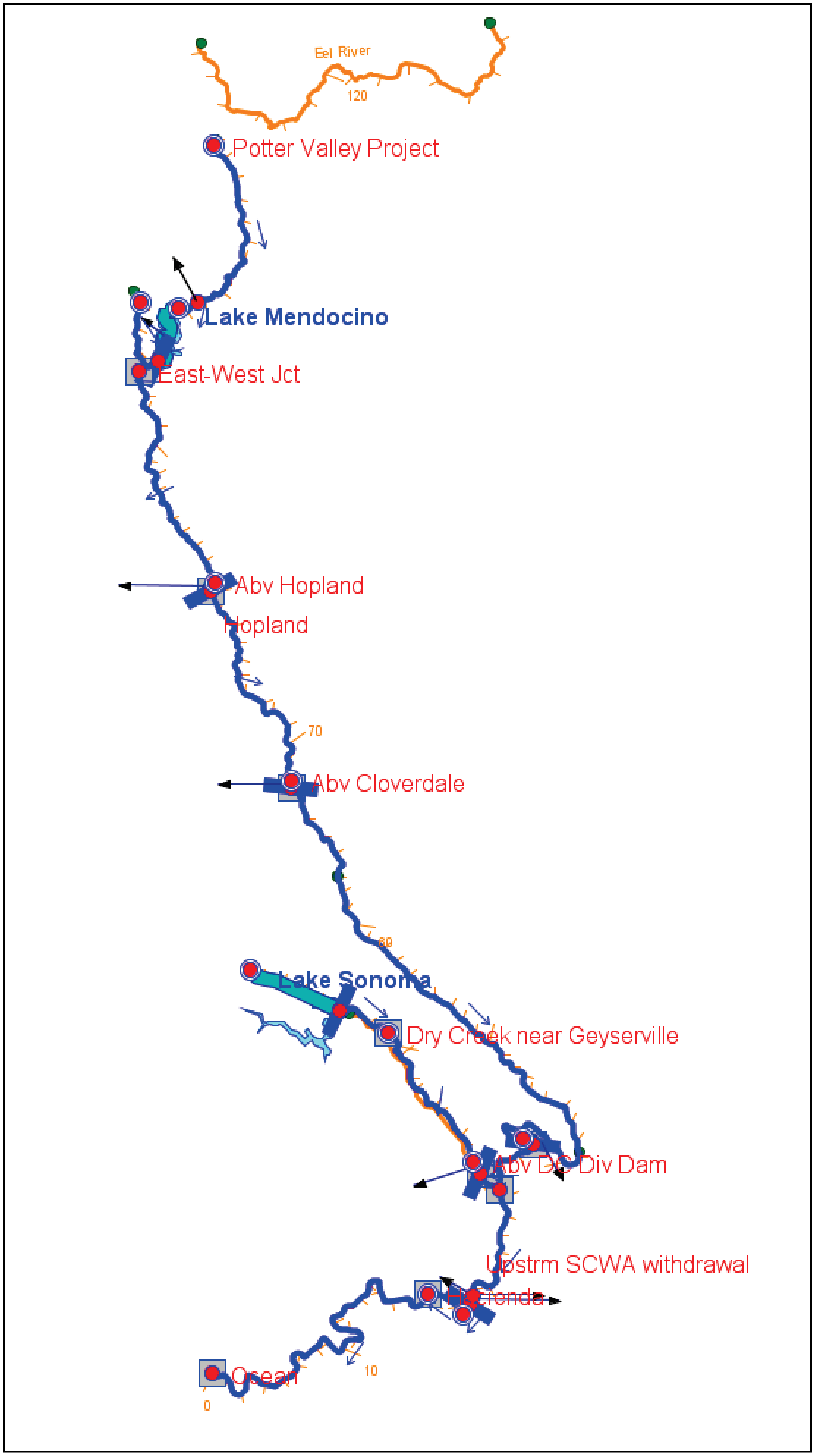

Figure 2. Schematic of Russian River ResSim Model.

Table 3. List of Junctions in the Russian River ResSim Model.

#	Junction Name	Junction Location	USGS Gage	Junction Gains	Junction Losses
1	Potter Valley Project	Tailrace Potter Valley Project	11461400	Eel River Model diversions	-
2	Coyote	Calpella Gage	11461500	-	Coyote
3	Lake Mendocino	Coyote Valley Dam	11462000	USGS Unimpaired Flows	Redwood Valley
4	West Fork Inflow	West Fork Russian River		USGS Unimpaired Flows	
5	East-West Jct	East and West Fork RR Confluence	-	-	-
6	Hopland	Hopland Gage	11462500	USGS Unimpaired Flows	Hopland
7	Cloverdale	Cloverdale Gage	11463000	USGS Unimpaired Flows	Cloverdale
8	Healdsburg	Healdsburg Gage	11464000	USGS Unimpaired Flows	Middle River
9	Lake Sonoma	Warm Springs Dam	11465000	USGS Unimpaired Flows	-
10	Dry Creek	Dry Cr nr Mouth Gage	11465350	USGS Unimpaired Flows	Dry Creek
11	Dry Creek - Russian River Jct	RR and Dry Cr Confluence	-	-	-
12	SCWA Diversion	Mirabel Inflatable Dam	-	-	SCWA, RRC, Lower River
13	Mark West Creek Jct.	RR and Mark W. Cr Confluence	-	USGS Unimpaired Flows	-
14	Hacienda	RR nr Guerneville Gage	11467000	USGS Unimpaired Flows	-

fourteen primary model junctions. Unimpaired flows from the USGS model and losses are added or removed at these junctions, as indicated in Table 3.

The model calculates flows and reservoir storage on a daily time-step. Reaches route water from one junction to another in the network. Routing is performed in HEC-ResSim using one of a handful of hydrologic routing methods. In this model, water was routed from upstream to downstream using a constant lag approach. Table 4 contains lag times for reaches with lag times greater than one day (the model simulation time-step was one day). These lag times were calibrated by SCWA for low flows, which were the focus of this study. These lag times would need to be adjusted if the model were to be used to evaluate flood events.

Table 4. Routing Parameters in the Russian River ResSim Model for Low Flows.

#	Reach Name	Lag Time (days)
1	Hopland to Cloverdale	2
2	Cloverdale to Healdsburg	3
3	Dry Creek to SCWA Diversion	2
4	Mark West to Hacienda	2
5	Lake Sonoma to Dry Creek – Russian River Jct	2

As described above, the system loss estimates were developed by SCWA. These were entered into the HEC-ResSim model as an annual time-series (disaggregated to a daily time-step). "Dummy" reservoirs were created in the model in order to link these loss time-series to the model so that water could be removed from the system. Figure 3 shows a plot of the nine different loss time-series.

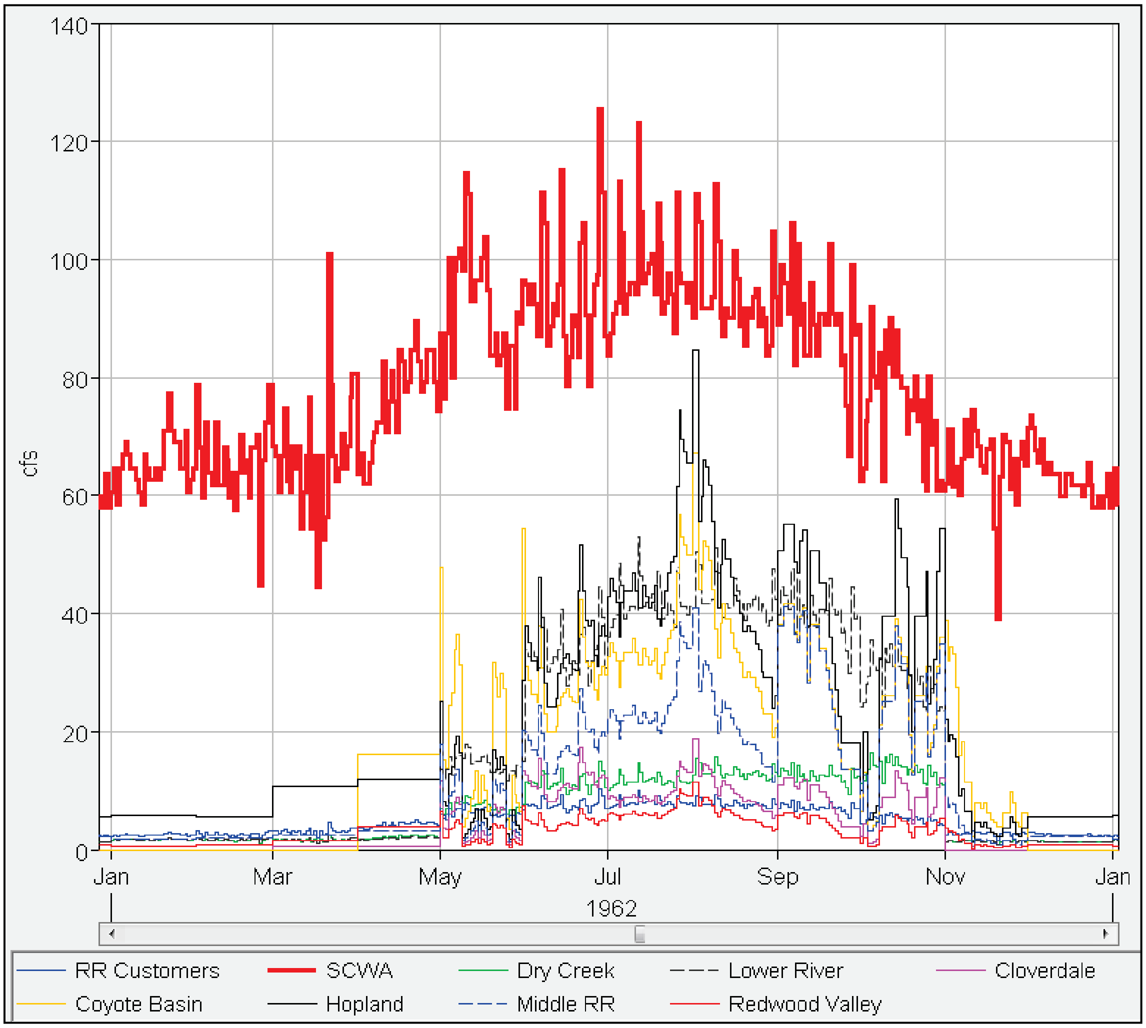

Figure 3. Losses Specified as Daily Time-Series (Annual Patterns).

The reservoir element in a HEC-ResSim model requires physical information as well as information describing operations. Physical data are represented by a pool and a dam. The pool contains the reservoir's elevation-storage relationship along with evaporation and seepage losses. The dam represents both uncontrolled (dam top and spillway) and controlled outlets. The operational data includes how the dam is operated to meet reservoir and channel constraints. The following sections contain the physical and operational data for Lake Mendocino and Lake Sonoma that was used in the HEC-ResSim model.

User-defined state variables were used to compute the hydrologic index. The state variables were developed using the built-in scripting interface in HEC-ResSim (all hydrologic index scripts are contained in Appendix A). The scripts were used to compute the hydrologic index based on model and external variables. For example, one hydrologic index alternative evaluates the storage in Lake Mendocino to determine the hydrologic index. In this case, the script was used to determine the current storage in Lake Mendocino and compare this storage to a threshold (defined by the study) to determine the hydrologic index. The hydrologic index is saved by the program and then used by the reservoir elements to set the minimum flow. This analysis is done by the program for each simulation time-step.

3.2 Lake Mendocino

Coyote Valley Dam was constructed in 1959 on the East Fork Russian River, creating Lake Mendocino. Coyote Valley Dam is an earthen embankment dam approximately 160 feet high with a crest length of 3,500 feet. Elevation of the dam crest is 784.0 feet. The approximately 120,000 acre-feet capacity (at spillway invert) of Lake Mendocino is used for flood control and flood conservation in the Russian River watershed. The outlet works consists of a single reinforced concrete pipe 11'-10" in diameter that feeds an eighteen-inch gate valve and a 108-inch cone valve for power generation. There are three sets of rectangular 5' x 9' service gates. The gates may be operated singly or in combination for flood releases. Operation of the outlet structure requires that if flows exceed 2,500 cfs at Ukiah, releases will be reduced to 25 cfs (if possible). During larger floods, releases will be minimized when flows are greater than 8,000 cfs at Hopland; therefore, the reservoir historically does not make releases during flood events.

Physical data for Lake Mendocino was found in the Water Control Manual (USACE, 1986). Reservoir storage capacity for Lake Mendocino is estimated to be 68,400 acre-feet at a normal pool elevation of 737.5 feet, 116,500 acre-feet at the spillway invert elevation of 764.8 feet, and 147,900 acre-feet at the spillway design flood elevation of 781.1 feet. These storage values are based on an updated bathymetric survey and do not match values in the Water Control Manual. The elevation-capacity curve developed using the most recent bathymetric survey data is shown in Figure 4. Uncontrolled and controlled releases were found in the Water Control Manual and are contained in Table 5. Monthly evaporation losses from Lake Mendocino are contained in Table 6.

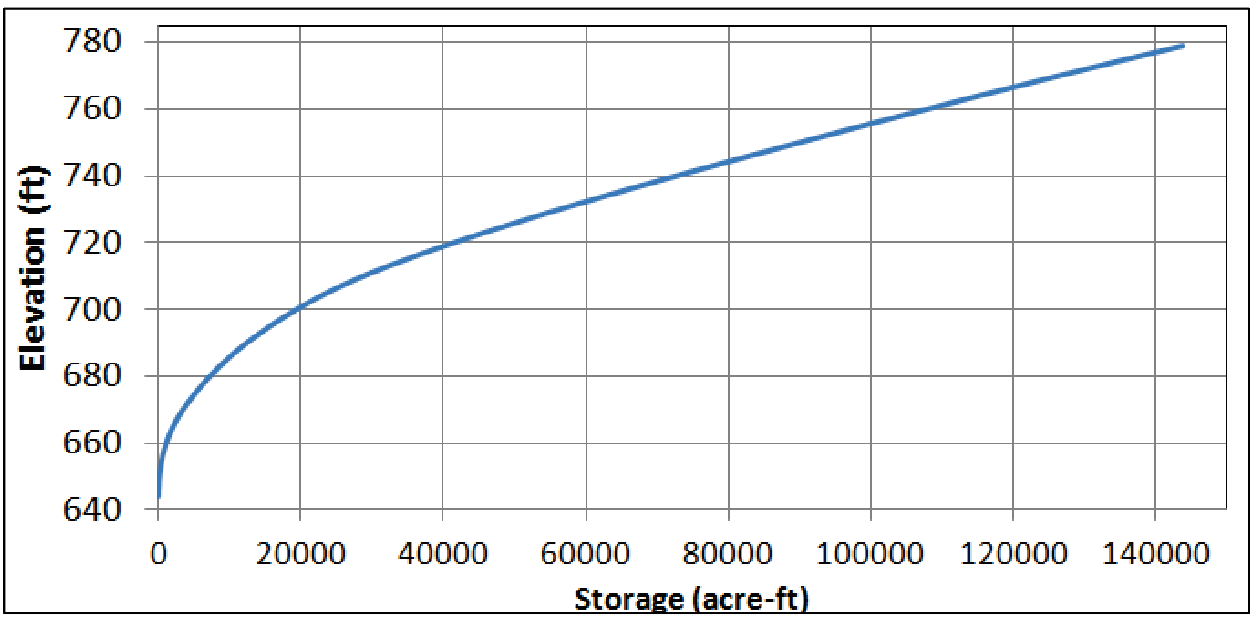

Figure 4. Elevation-Storage Curve for Lake Mendocino.

Operation of Lake Mendocino is described in the updated Water Control Manual (USACE, 2003) and includes operations for both flood control and water supply. Within HEC-ResSim, the reservoir element holds the operational data, which include rules for determining reservoir releases. The operational data is grouped as an operation set, and a reservoir can hold multiple operation sets. For this study, a separate operation set was created for each hydrologic index alternative. The operation set is made up of a set of operating zones, each of which contains a prioritized set of rules. Rules describe minimum or maximum reservoir releases, which can be based on a number of factors, such as downstream flow and current

Table 5. Controlled and Uncontrolled Releases for Coyote Valley Dam.

Elevation ft	Controlled Discharge cfs	Uncontrolled Discharge cfs	Total Discharge cfs
641	1,000	0	1,000
660	2,900	0	2,900
664	7,900	0	7,900
665	9,150	0	9,150
680	10,050	0	10,050
700	11,150	0	11,150
720	12,150	0	12,150
740	12,850	0	12,850
760	13,550	0	13,550
765	13,850	0	13,850
768	14,000	2,200	16,200
769	14,050	3,300	17,350
770	14,100	4,600	18,700
771	14,150	6,200	20,350
772	14,200	8,600	22,800
773	14,250	10,600	24,850
775	14,350	15,800	30,150
778	14,500	25,000	39,500
779	14,550	28,500	43,050
780	14,600	32,000	46,600
781	14,600	35,500	50,100
784	14,600	47,300	61,900

Table 6. Monthly Evaporation for Lake Mendocino.

Month	Evaporation (inches)
Jan	0.83
Feb	0.85
Mar	1.43
Apr	2.23
May	3.51
Jun	5.12
Jul	6.41
Aug	6.17
Sep	5.98
Oct	4.00
Nov	1.74
Dec	0.92

reservoir storage. Figure 5 shows the four storage zones for Lake Mendocino. The guide curve, also referred to as the target pool elevation, was designated manually in the model. Storage below the guide curve is referred to as conservation storage, and storage above the guide curve is referred to as flood control storage. HEC-ResSim determines releases from the reservoir based on where the current pool elevation is in relation to the guide curve. When the pool elevation is below the guide curve, the program reduces releases in order to fill the reservoir, and when the pool is above the guide curve the program makes releases to draw down the pool. Constraints (rules) defined by the modeler are applied when the program attempts to lower or raise the pool elevation to the guide curve.

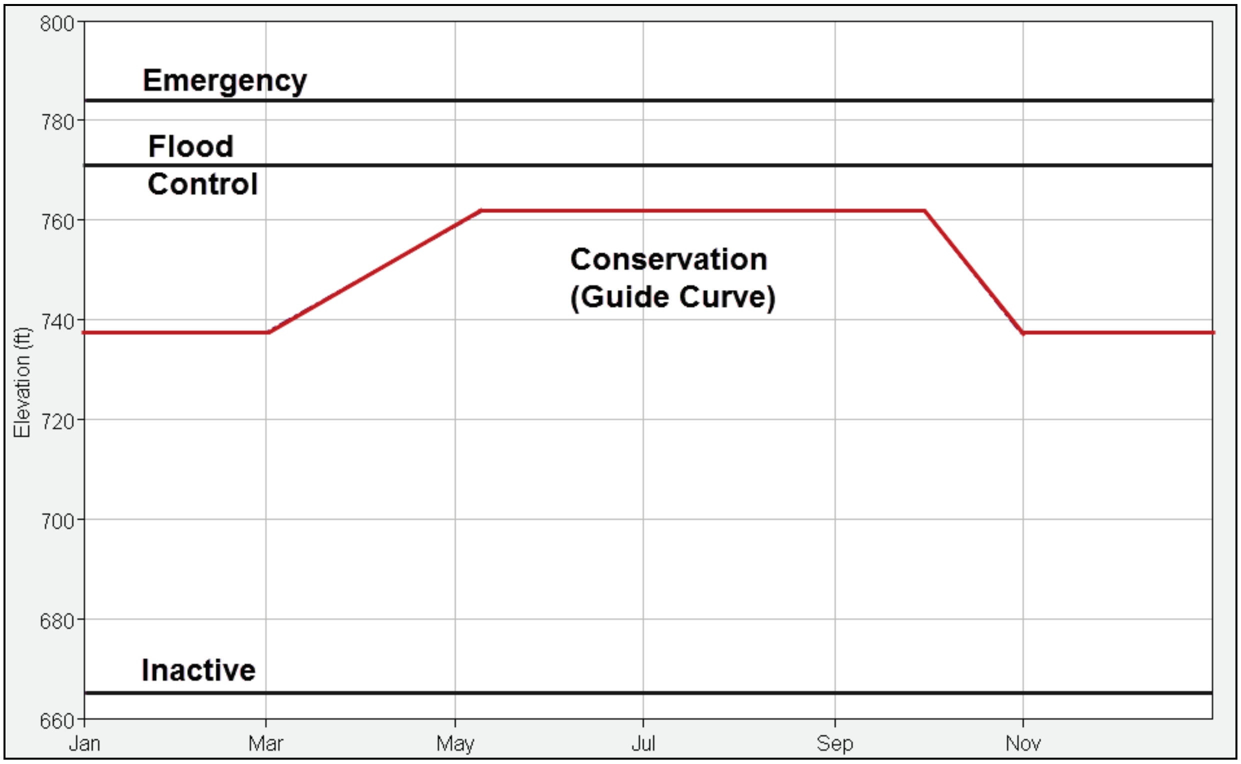

Figure 5. Lake Mendocino Seasonal Guide Curve.

The guide curve for Lake Mendocino varies throughout the year. From January to March 1 it is set to 737.5 feet (68,400 acre-feet), and from May 10 through September 30, it is 761.8 feet (110,000 acre-feet). The guide curve is lowered in October back to 737.5 feet, since the flood season in the Russian River watershed typically extends between November through the end of March. Lowering the guide curve in these months provides added flood protection (41,600 acre-feet). The guide curve begins increasing in March because 1) it becomes less likely for large rainfall-runoff event to occur in the Russian River watershed; and, 2) it is important to capture (store) any late season runoff before the drier summer and fall months.

Table 7 contains the operation set for Lake Mendocino using the current hydrologic index. The operation sets for the other hydrologic index alternatives are very similar to the current index operation set. The only difference is in the WSC I-1610 Q-BO "If" block. Some of the new hydrologic index alternatives only evaluate storage in Lake Mendocino, instead of the combined storage in Lakes Mendocino and Pillsbury.

3.3 Lake Sonoma

Warm Springs Dam was completed in 1982 on Dry Creek, creating Lake Sonoma. Warm Springs Dam is a compacted earth fill with impervious core dam, with a maximum height above the streambed of 319 feet and a crest length of 3,000 feet. Elevation of the dam crest is 519.0 feet. The approximately 381,000 acre-feet capacity (at spillway invert) of Lake Sonoma is used for flood control and flood conservation in the Russian River basin. The outlet works consist of a low flow water quality outlet with three five-foot diameter intake tunnels (at elevations 431, 391, and 352 feet). The flood control outlets consist of two 5' x 8' service gates and two 5' x 8' slide gates. Outlets are operated to restrict flows to 7,000 cfs at the Yokum Bridge near Geyserville and to not exceed 35,000 cfs on the Russian River at Guerneville.

Table 7. Operation Set for Lake Mendocino for the Current Hydrologic Index.

Lake Mendocino	Description
Emergency Zone	**Above 771 feet to the top of the dam, 784 feet**
MaxReleaseFlood_Gates	Limit flow through controlled and uncontrolled outlet to less than 8,000 cfs until pool is above 773 feet.
Flood Control Zone	**Above guide curve to 771 feet**
RVWD Diversion	"If" logic to limit the amount of diversion to the Redwood Valley Diversion (function of Lake Mendocino Storage).
Min25-Release	Sets the minimum release from the dam to 25 cfs for all simulation time-steps.
WSC I-1610 Q-BO	"If" logic that sets the minimum flow on the upper reach, at the confluence of East and West Forks of the Russian River, Hopland, Cloverdale, and Healdsburg, to minimum flows defined by the hydrologic index. The current hydrologic index looks at cumulative flow into Lake Pillsbury and storage at both Lakes Pillsbury and Mendocino (state variable is used to set the hydrologic index). Minimum releases are set to those shown in Figure 1.
RR_Ukiah_Flow	"If" logic used to set releases to 25 cfs if flows on the West Fork exceed 2,500 cfs.
MaxRealeaseWCM-FC	Used to set maximum flows from the dam, based on elevation (737.5 ft 2000 cfs, 746.0 ft 4000 cfs, 755.0 ft 6400 cfs, 771.0 ft 6400 cfs).
MaxHopland	Used to set maximum releases so that flow at Hopland does not exceed 8,000 cfs.
Conservation Zone	**Above 665 feet to the guide curve**
RVWD Diversion	Same as above
Min25-Release	Same as above
WSC I-1610 Q-BO	Same as above
RR_Ukiah_Flow	Same as above
MaxHopland	Same as above
Inactive	**665 feet**

Physical data for Lake Sonoma was found in the Water Control Manual (USACE, 2003). Reservoir storage capacity for Lake Sonoma is estimated to be 245,000 acre-feet at a normal pool elevation of 451.1 feet, 381,000 acre-feet at the spillway invert elevation of 495.0 feet, and 449,000 acre-feet at the spillway design flood elevation of 513.1 feet. The elevation-capacity curve is shown in Figure 6. Uncontrolled and controlled releases were found in the Water Control Manual and are contained in Table 8. Monthly evaporation losses from Lake Sonoma are contained in Table 9.

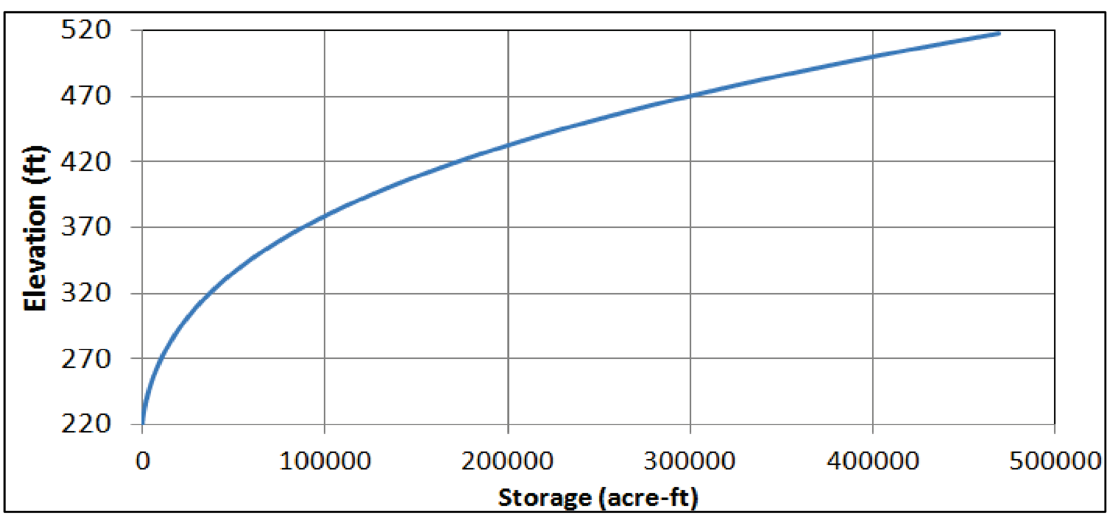

Figure 6. Elevation-Storage Curve for Lake Sonoma.

Table 8. Controlled and Uncontrolled Releases for Warm Springs Dam.

Elevation ft	Controlled Discharge cfs	Uncontrolled Discharge cfs	Total Discharge cfs
221.0	0	0	0
250.0	2,478	0	2,478
292.0	4,186	0	4,186
300.0	4,490	0	4,490
350.0	5,710	0	5,710
352.0	5,749	0	5,749
354.0	5,788	0	5,788
400.0	6,690	0	6,690
480.0	7,757	0	7,757
495.0	7,957	0	7,957
495.8	7,967	400	8,367
496.7	7,979	1,000	8,979
497.7	7,993	1,750	9,743
500.0	8,023	3,820	11,843
500.2	8,026	4,000	12,026
501.7	8,046	5,800	13,846
504.2	8,079	9,500	17,579
505.0	8,090	11,017	19,107
507.1	8,090	15,000	23,090
510.0	8,090	22,176	30,266
513.0	8,090	29,600	37,690
519.0	8,090	45,000	53,090

Table 9. Monthly Evaporation for Lake Sonoma.

Month	Evaporation (inches)
Jan	0.79
Feb	0.97
Mar	1.68
Apr	2.68
May	4.00
Jun	5.11
Jul	6.46
Aug	6.40
Sep	5.92
Oct	4.42
Nov	2.30
Dec	1.01

Operation of Lake Sonoma is described in the updated Water Control Manual (USACE, 2003) and includes operations for both flood control and water supply. Within HEC-ResSim, the reservoir element holds the operational data, which include rules for determining reservoir releases. The operational data is grouped as an operation set, and a reservoir can hold multiple operation sets. For this study, a separate operation set was created for each hydrologic index alternative. The operation set is made up of a set of operating zones, each of which contains a prioritized set of rules. Rules describe minimum or maximum reservoir releases, which can be based on a number of factors, such as downstream flow and current reservoir storage. Figure 7 shows the four storage zones for Lake Sonoma. The guide curve was designated manually in the model. Storage below the guide curve is referred to as conservation storage, and storage above the guide curve is referred to as flood control

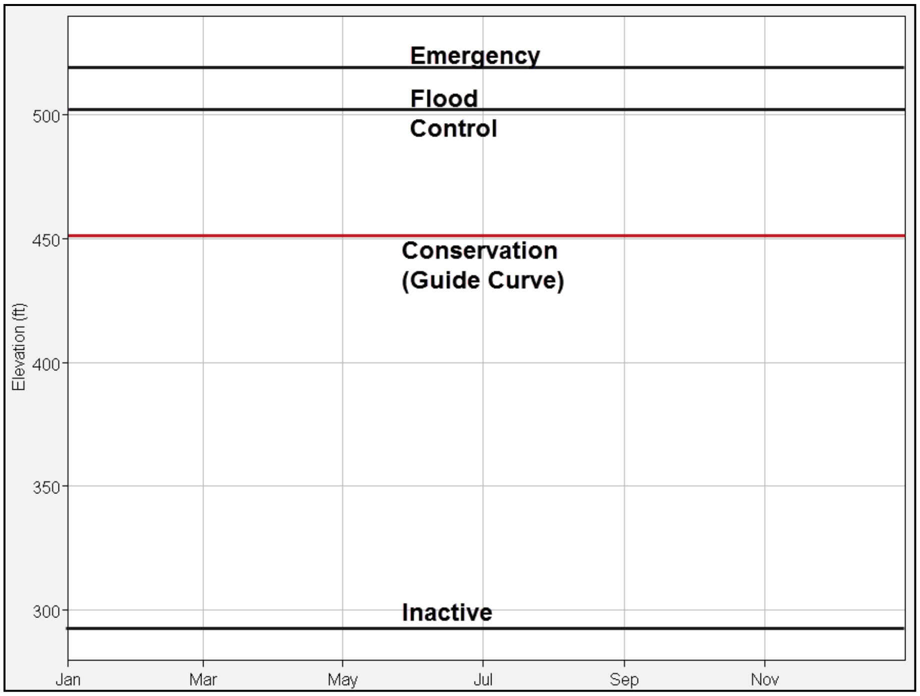

Figure 7. Lake Sonoma Guide Curve.

storage. HEC-ResSim determines releases from the reservoir based on where the current pool elevation is in relation to the guide curve. When the pool elevation is below the guide curve, the program reduces releases in order to fill the reservoir, and when the pool is above the guide curve the program makes releases to draw down the pool. Constraints (rules) defined by the modeler are applied when the program attempts to lower or raise the pool elevation to the guide curve. Table 10 contains the operation set for Lake Mendocino using the current hydrologic index. The operation sets for the other hydrologic index alternatives are very similar to the current index operation set. The only difference is in the WSC I-1610 Q-BO "If" block.

The SCWA's water rights permits include a provision that requires SCWA to impose a thirty percent deficiency in deliveries from the Russian River to its service area when Lake Sonoma storage levels drop below 100,000 acre-feet before July 15 of any year. According to the SWCA's water rights permits this deficiency must remain in effect until "(1) storage in Lake Sonoma rises to greater than 70,000 acre-feet subsequent to December 31 after having fallen below that level, or (2) permittee has projected, to the satisfaction of the Chief, Division of Water Rights, that storage at Lake Sonoma will not fall below 70,000 acre-feet , or (3) hydrologic conditions result in sufficient flow to satisfy permittee's demands at Wohler and Mirabel Park and minimum flow requirements in the Russian River at Guerneville". This provision is intended to ensure the maintenance of minimum stream flows required by Decision 1610. This provision is accounted for in the modeling, although the model assumes delivery deficiencies remain in effect at least until storage has recovered in Lake Sonoma to greater than 70,000 acre-feet after December 31. The model does not allow for earlier termination of deficiencies based on hydrologic conditions.

3.4 HEC Review of the Russian River ResSim Model

The Russian River HEC-ResSim model was originally developed by a contractor for SCWA. Before the model was used for this study, it was reviewed by CEIWR-HEC and changes were

Table 10. Operation Set for Lake Sonoma for the Current Hydrologic Index.

Lake Sonoma	Description
Emergency Zone	**Above 502 feet to the top of the dam, 519 feet**
MaxReleaseFlood_Gates	Limit flow through controlled and uncontrolled outlet to less than 8,000 cfs until pool is above 505 feet
Flood Control Zone	**Above guide curve to 502 feet**
WSC I-1610 Q-BO	"If" logic that sets the minimum flow on Dry Creek and the lower Russian River to minimum flows defined by the hydrologic index. The current hydrologic index looks at cumulative flow into Lake Pillsbury and storage at both Lakes Pillsbury and Mendocino (state variable is used to set the hydrologic index). Minimum releases are set to those shown in Figure 1.
MaxGeyserville	Used to set maximum releases so that flow on Dry Creek near Geyserville does not exceed 7,000 cfs
MaxHacienda	Used to set maximum releases so that flow on the Russian River at Hacienda Bridge does not exceed 35,000 cfs
rising	"If" logic used to set releases from the dam to 25 cfs if inflow into the dam is increasing and above a threshold or set releases based on reservoir elevation (USACE, 1984).
Conservation Zone	**Above 292.7 feet to the guide curve**
WSC I-1610 Q-BO	Same as above
MaxGeyserville	Same as above
MaxHacienda	Same as above
rising	Same as above
Inactive	**292.7 feet**

made to improve flood control operations and decrease model runtimes (modify storage curves for diversion dams and modify state varibles). The model was modified so that flood operations could be evaluated. All diversions were removed, and the reservoirs were forced to make minimum releases (observed discharge) from approximately April through November (outside the flood season). Local inflows were computed using observed flow (observed flow from an upstream gage was routed to the downstream gage and subtracted from the observed flow). This was done to create a flow dataset that could be used to recreate observed flows at gage locations. This dataset allowed the analysis to focus on reservoir operation and not modeling of local runoff. Inflows into Lake Mendocino were computed using measured flow at Calpella and an adjustment for drainage area. Inflow into Lake Sonoma was computed using observed outflow and storage. This includes inflows from 1992 – 2008. For 1981 – 1992 the model uses inflows computed by the USGS model.

The following list contains changes made to the Russian River ResSim model to improve flood operation and to match operations contained in the water control manuals.

Model changes:

- The lag time for routing reaches was modified since the lag is much less during high flows. The lag time was determined by evaluating the time of peak flows from large flood events at upstream and downstream gages.
- The Dry Creek near Geyserville junction was added to the model because Lake Sonoma operates for this location.
- Monthly evaporation was updated using data in the reservoir control manuals (originally, data for Lake Sonoma was considerably higher than the manual). The model uses one annual pattern to model historic evaporation.
- Physical properties of the dam were updated to match the Reservoir Control Manuals.
- The Controlled and Uncontrolled Outlets (gated outlets and spillways) were updated for both reservoirs to reflect data in the Reservoir Control Manuals.

- The Zones for Lake Mendocino were modified, Flood Zones 1 – 3 were merged into one flood zone. A rule was created controlling releases as a function of pool elevation (this is used to model discharge once the pool was filled into one of the flood zones).
- An "emergency zone" was added to both reservoirs. This zone is included in the Reservoir Control Manuals and controls releases when the reservoir pool is above the spillway invert. This is used to control releases from gated control structures during spillway discharge.
- In Lake Sonoma, a new rule was added to operate for a maximum flow at the Dry Creek near Geyserville gage. Flow rates in the increasing and decreasing rate of change rules were modified based on observed releases.
- In Lake Mendocino, a new rule was added to monitor flow at the Russian River at Ukiah gage. If flow exceeds 2,500 cfs then releases from the Lake Mendocino are reduced to 25 cfs. Rate of change rules were also modified based on observed releases.

Figure 8 shows simulated and observed reservoir inflow and storage for Lake Mendocino during the 2006 water year. Results show the reservoir storing and releasing water so that flow at downstream operation points stays below maximum restraints. It is evident from

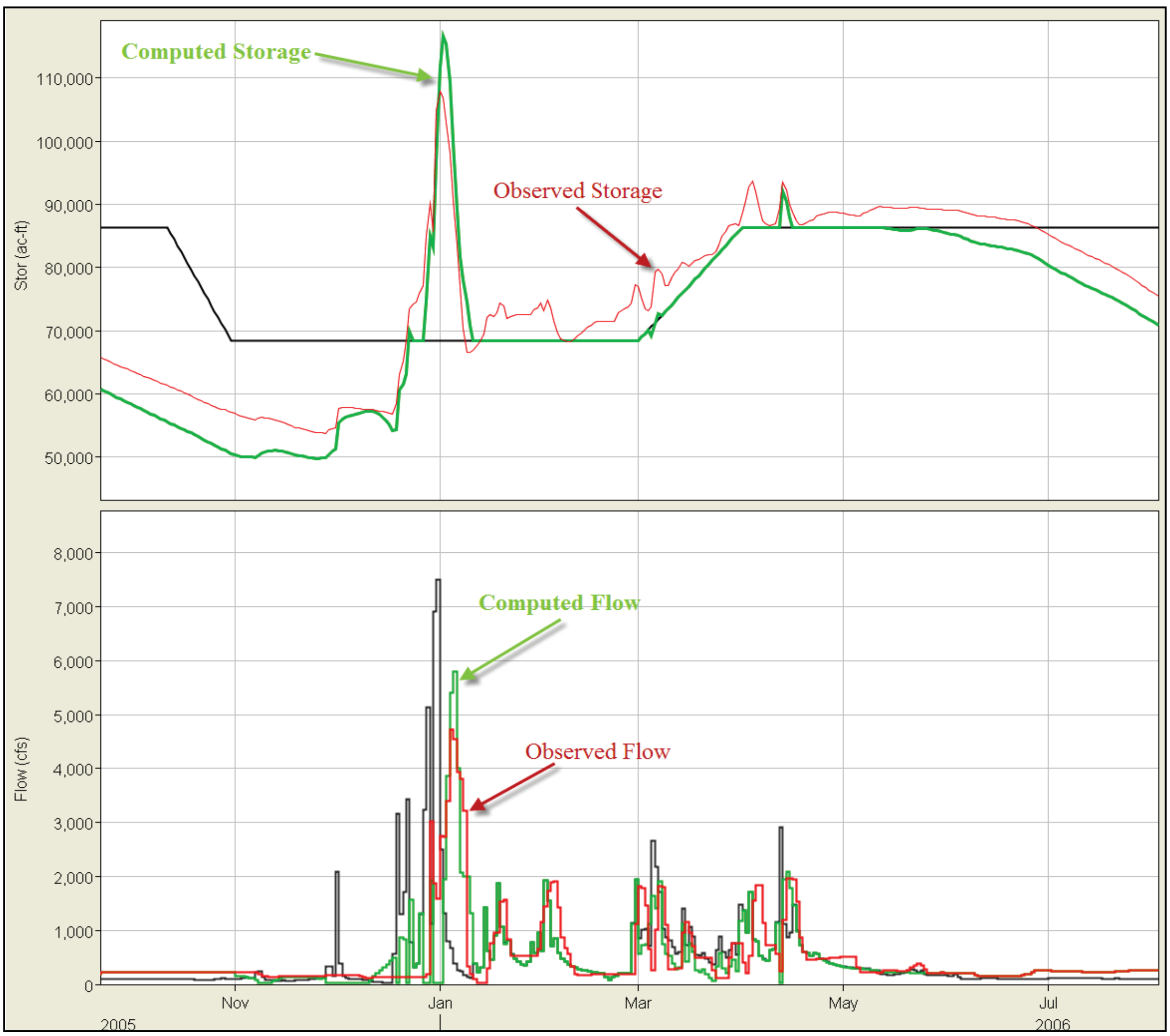

Figure 8. Comparison of Observed and Simulated Storage and Discharge from Lake Mendocino (for calibration purposes, the guide curve is from the 1986 WCM).

Figure 8 that the HEC-ResSim model operates to the guide curve while actual operation slightly deviates (encroaches into the flood control pool). Figure 8 also shows a guide curve in late spring and summer that is less than the current guide curve. The guide curve from the 1986 water control manual was used in order to calibrate the model to historic events (the current guide curve was not inacted until 2008). The current guide curve was used for all hydrologic index simulations. Figure 9 shows simulated and observed reservoir inflow and storage for Lake Sonoma during the 2006 water year. Simulated versus observed results show the reservoir storing and releasing water in a similar manner as the operations defined in the water control manunal.

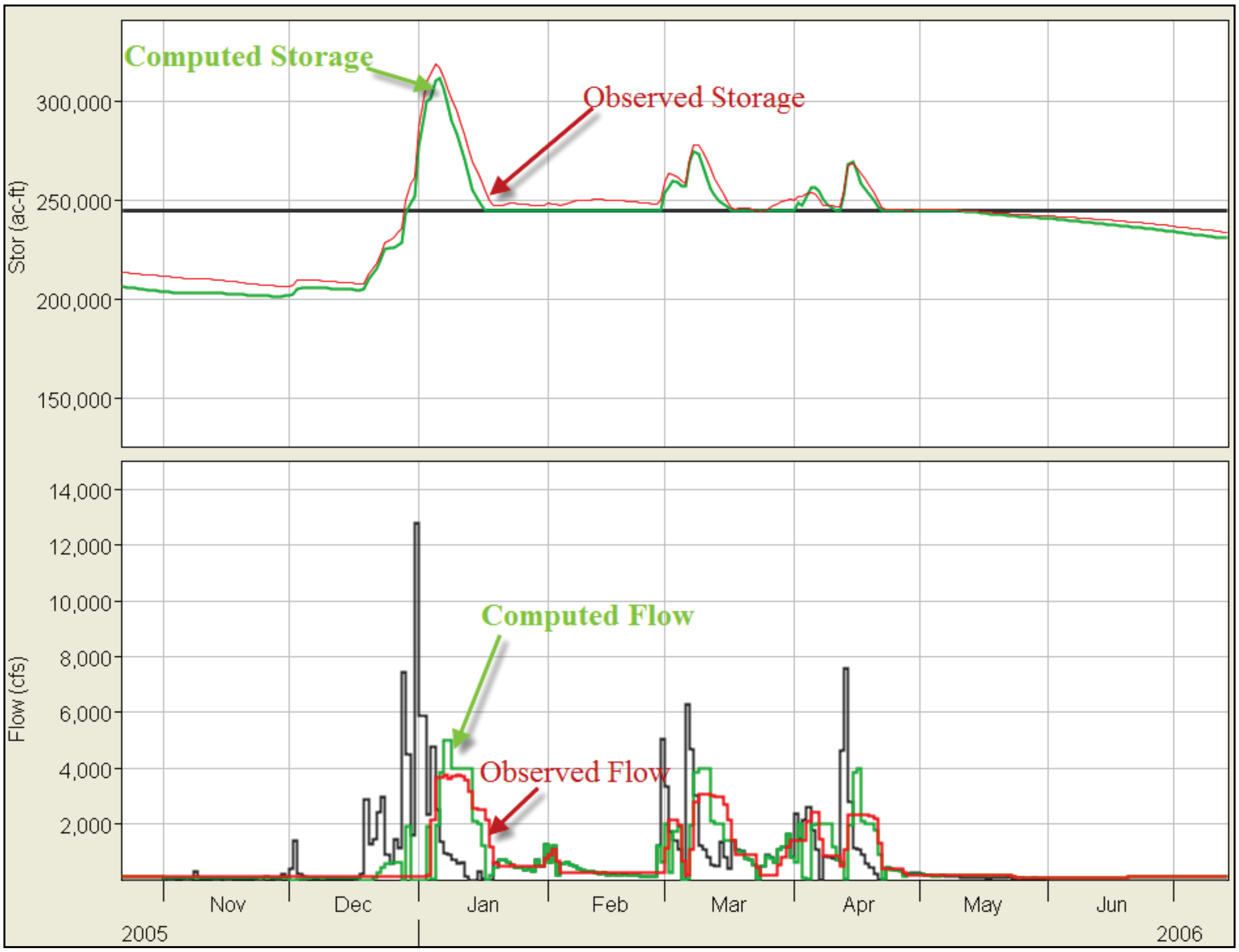

Figure 9. Comparison of Observed and Simulated Storage and Discharge from Lake Sonoma.

Chapter 4
Development of a New Hydrologic Index

The current methodology for determining the hydrologic index (and the minimum flow requirments) is defined in SCWA's water rights permits, which were issued following the State Water Resource Control Board Decision 1610. As previousely stated, this methodology uses cumulative inflow into Lake Pillsbury (not in the Russian River Watershed) starting on October 1 for each water year. Cumulative inflow beginning on October 1 is assessed on the first of the month starting January 1 and is continued through June 1 to determine the hydrologic index, Normal, Dry, or Critical. In addition, total storage in Lake Pillsbury and Lake Mendocino is evaluated on May 31. If the combined storage is less than 130,000 acre-feet , then a Normal index is changed to Dry for the Upper Russian River (Coyote Valley Dam to the confluence with Dry Creek). This check is used to catch those years when a majority of the rainfall-runoff occurred prior to March, when the guide curve is low, and when the reservoir was not able to store the runoff for water supply later in the season. Finally, a Normal index can be changed to Dry on any day from October 1 through December 31 if the storage in Lake Mendocino falls below 30,000 acre-feet . The hydrologic index determines the minimum instream flow requirements in the Russian River and Dry Creek (refer to Table 1 and Figure 1).

A time-series (1910-2008) of the current hydrologic index was created by SCWA using computed/observed inflows into Lake Pillsbury and storage (simulated) in both Lake Pillsbury and Lake Mendocino. This time-series only incorporates the cumulative inflow portion of the hydrologic index. The Lake Pillsbury inflow dataset is a combination of estimated flows using observed reservoir outflow, reservoir storage, and estimated evaporation losses from 1922 – 2008 and observed flows prior to 1922. Figure 10 shows the cumulative inflow into Lake Pillsbury and the hydrologic index computed using the current method during 1918. As Figure 10 shows, 1918 was a dry year; therefore, the

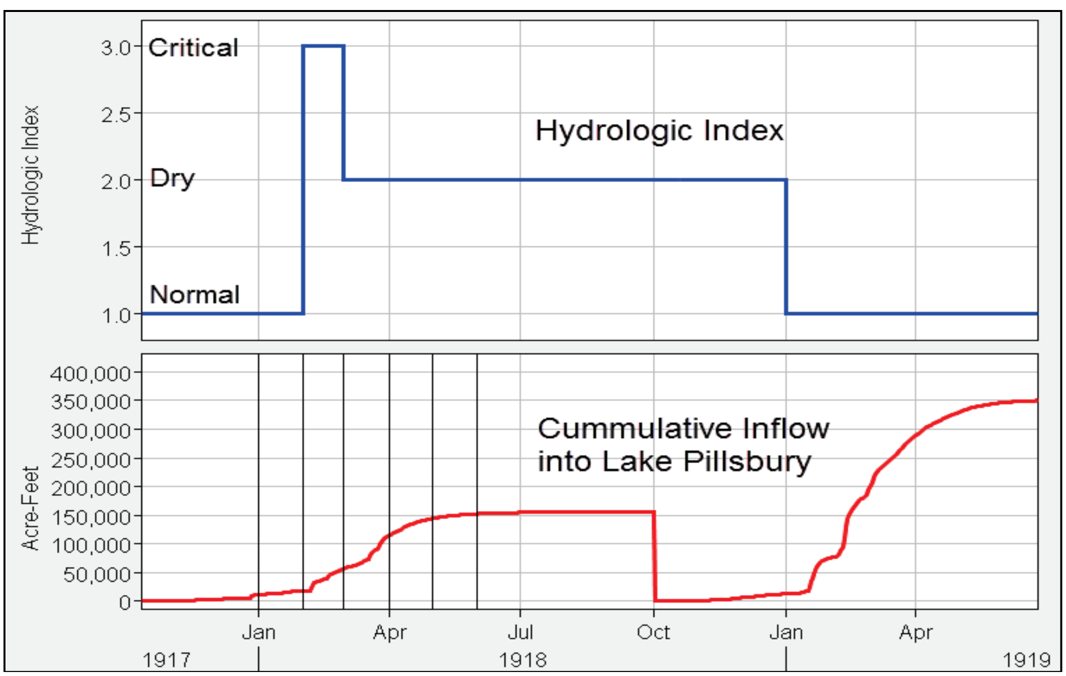

Figure 10. Cumulative Inflow into Lake Pillsbury and the Hydrologic Index Computed using the Current Method.

hydrologic index was Critical in February and Dry from March through December. Figure 11 shows the cumulative Lake Pillsbury inflow and hydrologic index for the entire period of record. The hydrologic index results presented in the figure does not include Dry Spring conditions as determined by combined Lake Pillsbury and Lake Mendocino storage on May 31 of each year.

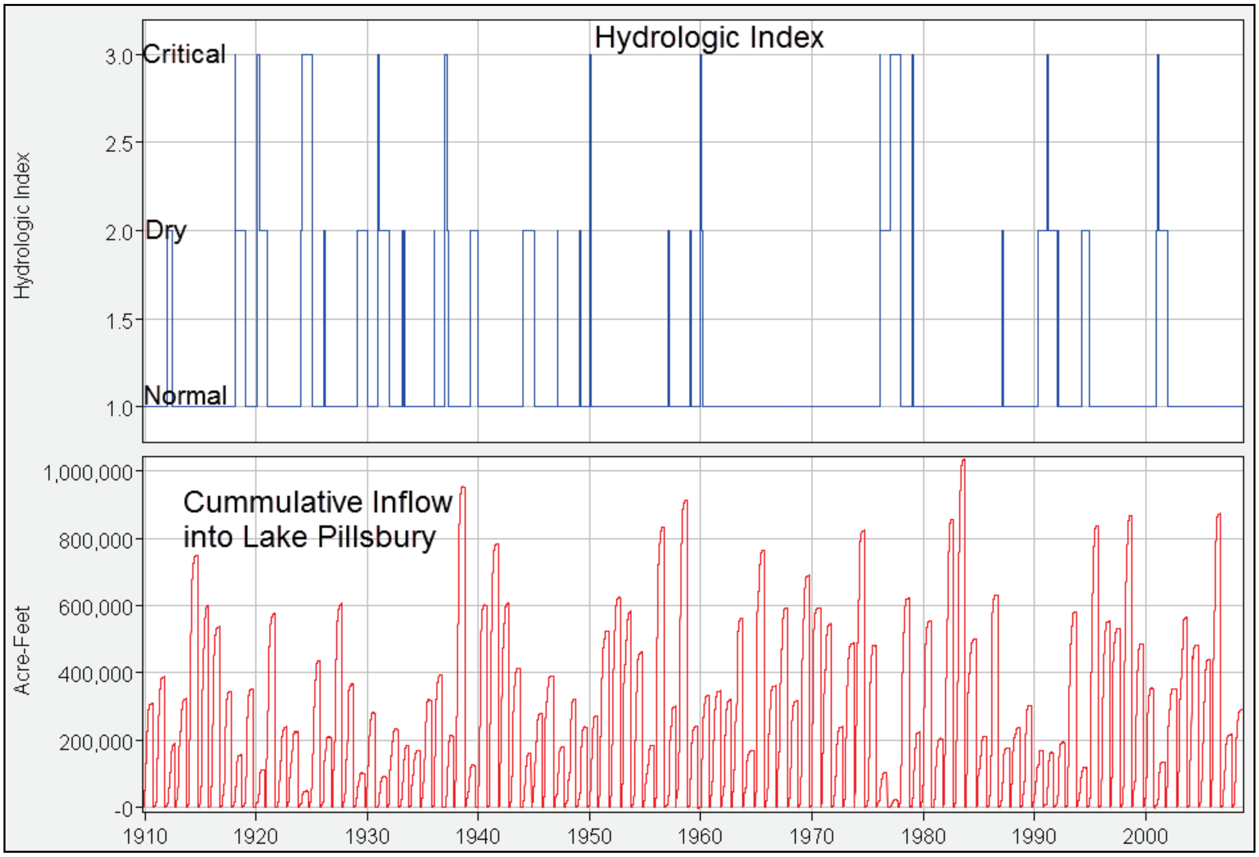

Figure 11. Period-of-Record Cumulative Inflow into Lake Pillsbury and the Hydrologic Index.

4.1 Base Analysis

A base analysis (using the HEC-ResSim model) was used to determine the percent of time that Normal, Dry, and Critical indices occur with the current hydrologic index criteria. The base analysis was simulated by incorporating all the boundary conditions previously mentioned (USGS unimpaired inflows, inflow from the PVP, and losses), reservoir operation, and the "historic" time-series of hydrologic index. This historic time-series was developed using the current hydrologic index with only cumulative inflow evaluated; storage in Lake Mendocino and Lake Pillsbury were not included as their affect on the hydrologic index could only be evaluated with a model simulation. As previousely stated, total storage in Lake Pillsbury and Lake Mendocino is evaluated on May 31. If the combined storage is less than 130,000 acre-feet, then a Normal index is changed to Dry. Also, a Normal index can be changed to Dry any day from October 1 through December 31 if the storage in Lake Mendocino falls below 30,000 acre-feet.

Output from an HEC-ResSim simulation includes a time-series (at a one-day simulation time-step) of the hydrologic index. This was processed to determine the percentages of Normal, Dry, and Critical periods during period-of-record simulation. Table 11 contains the monthly distribution of hydrologic index from the 98-year period-of-record (these values

Table 11. Percentage of Time Normal, Dry, and Critical Hydrologic Indices were Observed During the 98-year period-of-Record.

	Percentages		
	Normal	Dry	Critical
January	86%	8%	6%
February	79%	13%	8%
March	86%	10%	4%
April	87%	10%	3%
May	85%	13%	2%
June - December	76%	22%	2%
Annual Average	76.9%	20.0%	3.1%

include the dry spring and minimum Lake Mendocino conditions). This table shows that a Normal index was observed 76.9 percent of the time, a Dry index was observed twenty percent of the time, and a Critical index was observed 3.1 percent of the time. Figure 12 shows the combined storage in Lake Pillsbury and Lake Mendocino. There are a number of years where the combined storage on May 31 was less than 130,000 acre-feet (seventeen

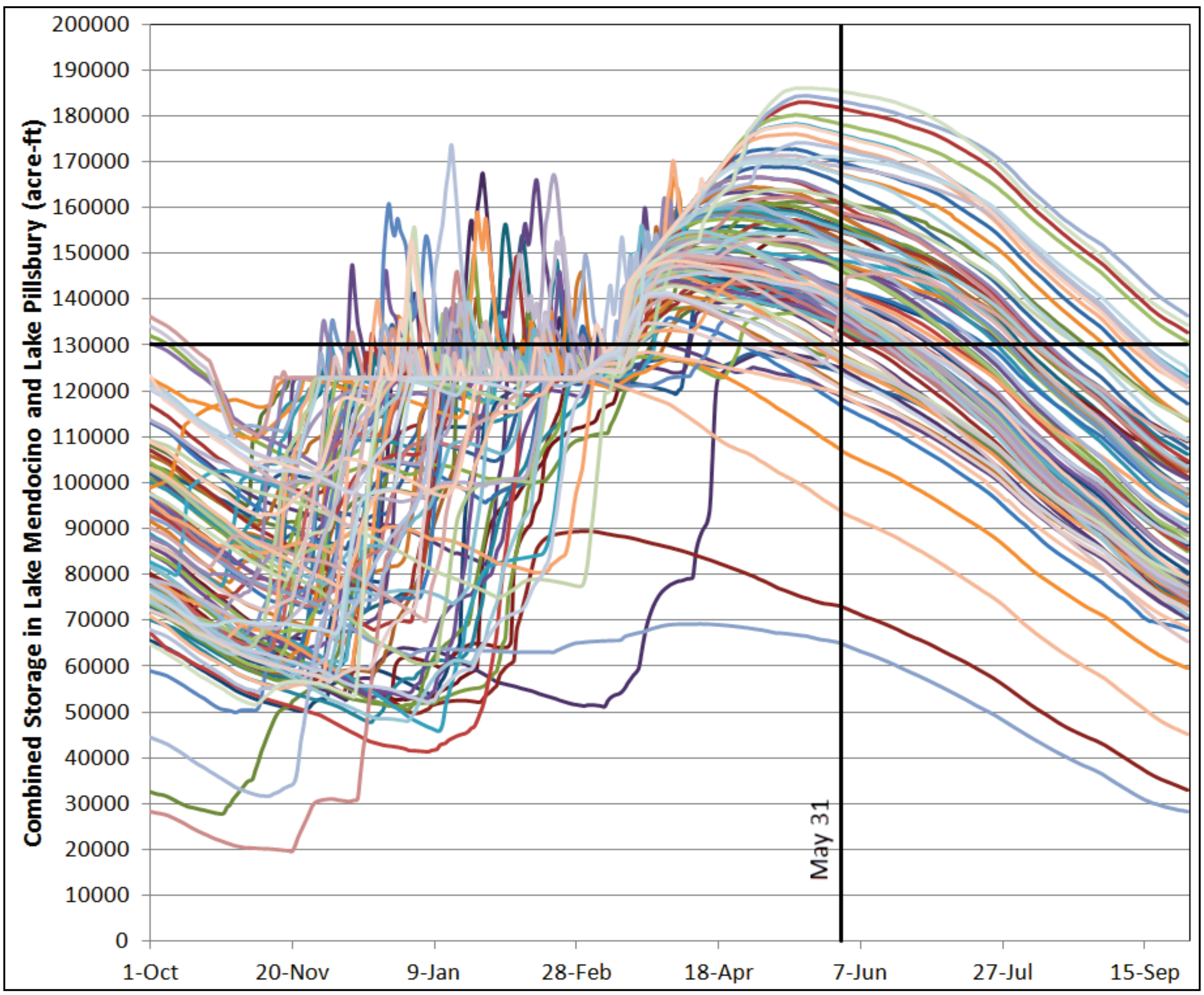

Figure 12. Combined Storage in Lake Mendocino and Lake Pillsbury.

water years out of 98 had a May 31 storage less than 130,000 acre-feet). In addition, results shows that storage in Lake Mendocino fell below 30,000 acre-feet 1,427 times from October 1 through December 31 during the 98-year period-of-record. If only cumulative inflow into Lake Pillsbury were evaluated, the percent of time each index would be observed is 85.7 percent for Normal, 11.2 percent for Dry, and 3.1 percent for Critical. This shows that the storage criterion (combined storage on May 31 and Lake Mendocino storage from October 1 through December 31) increases the percent of Dry periods from 11.2 to twenty percent during the 98-year period-of-record.

The "dry" spring condition only requires modification of minimum flows for the upper Russian River and not the entire system. Adjusting the Dry periods for dry spring conditions or minimum storage in Lake Mendocino is only relevant for the upper river.

The base analysis was also used to illustrate how changes in PVP water affected the current hydrologic index (percentage of Normal, Dry, and Critical periods) and storage in Lake Mendocino. As previously stated, diversions from the Eel River (PVP water) into the East Fork Russian River were computed for the 98-year period-of-record by SCWA to reflect current operations. A comparison of actual, "historic", PVP water and the simulated current PVP water is shown in Figure 13. Notice that historic average daily PVP diversions are approximately 200 cfs while current operation average daily diversions are approximately

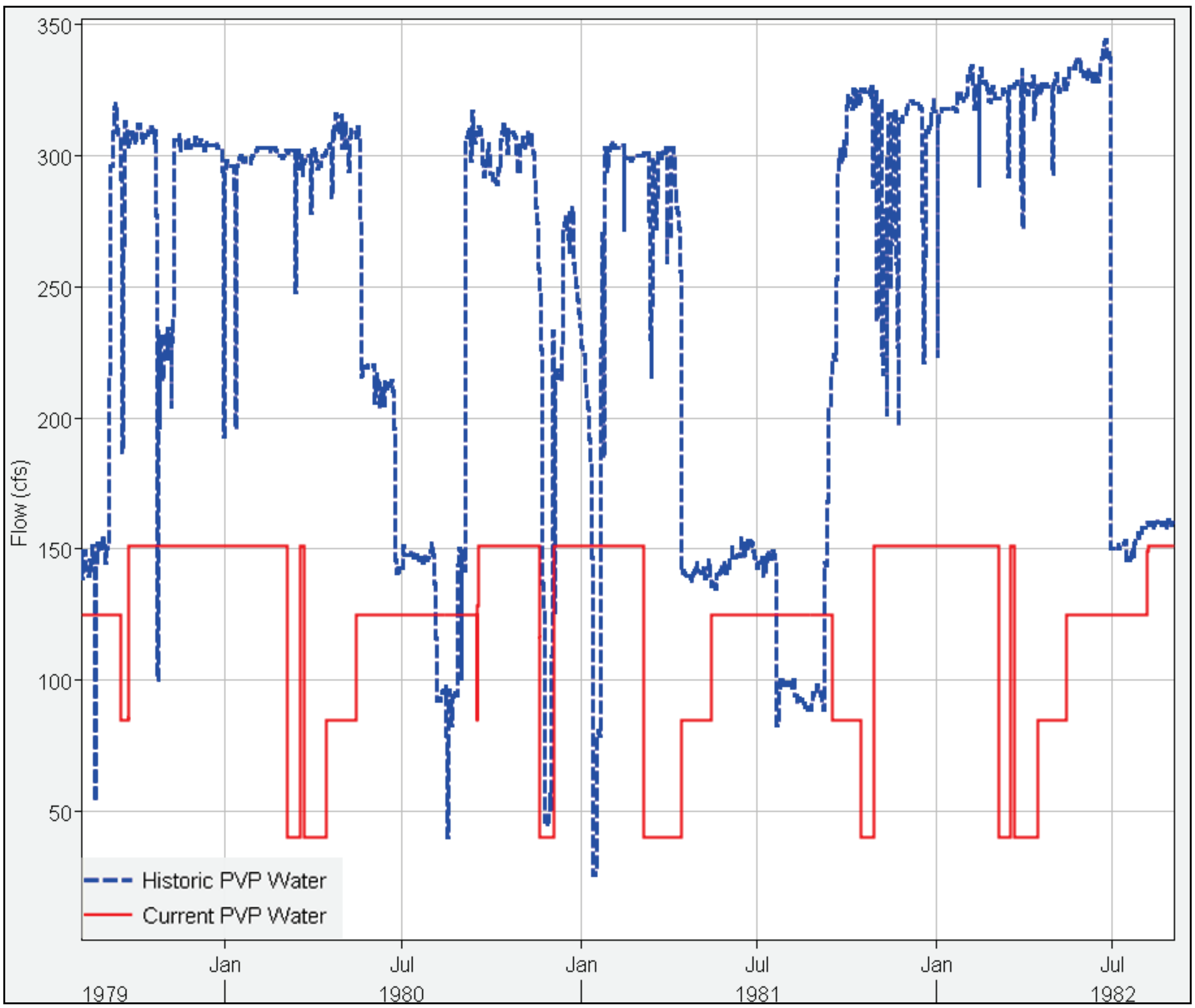

Figure 13. Current and Historic PVP Water.

105 cfs. A HEC-ResSim simulation was created using the historic PVP diversions. Figure 14 shows the simulated storage in Lake Mendocino for both the historic and current PVP diversion conditions. The current PVP diversions result in much lower storage in Lake

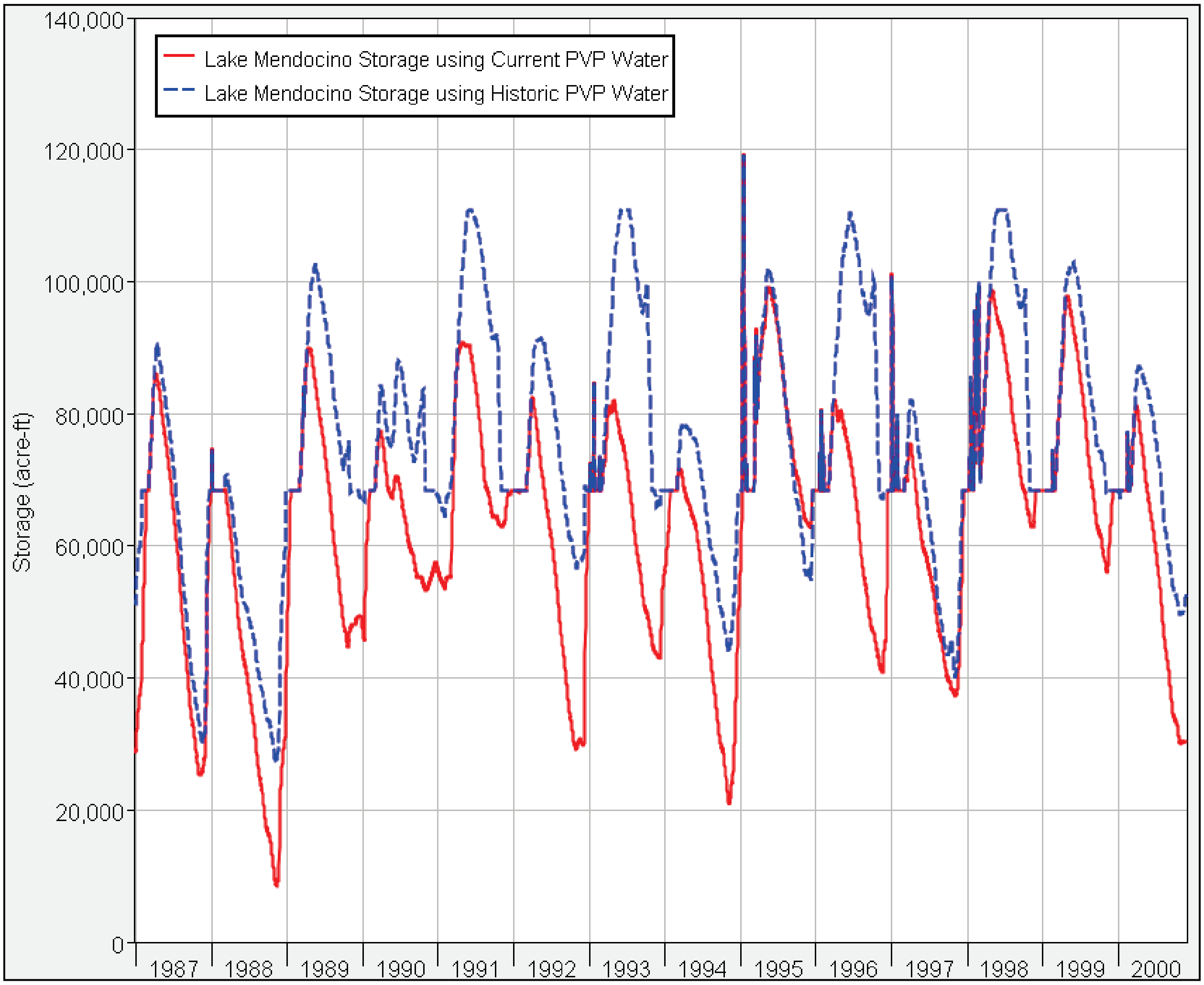

Figure 14. Lake Mendocino Storage using Current and Historic PVP Water.

Mendocino during summer and fall months. The percentages of Normal, Dry, and Critical are 83.5, 13.4, and 3.1 percent, respectively, when using the historic PVP water. These numbers can be compared to those in Table 11 to show that the reduction in PVP diversions results in a Dry hydrologic condition that is observed more often over the 98-year period-of-record (Dry index was observed twenty percent of the time with the current PVP water versus 13.4 percent of the time with the historic PVP water).

4.2 Lake Mendocino Cumulative Inflow Hydrologic Index

An alternative hydrologic index was developed by directly applying the current index approach to cumulative inflow into and storage in Lake Mendocino only. Figure 15 shows the cumulative inflow into Lake Mendocino during the 98-year period-of-record (the dark red line is the average from the 98 years, complete water years). This inflow dataset was computed using the USGS unimpaired inflow data, simulated PVP water, and losses. The cumulative inflows were analyzed and thresholds were set in order to match the same number of Normal, Dry, and Critical periods experienced from the Lake Pillsbury index when

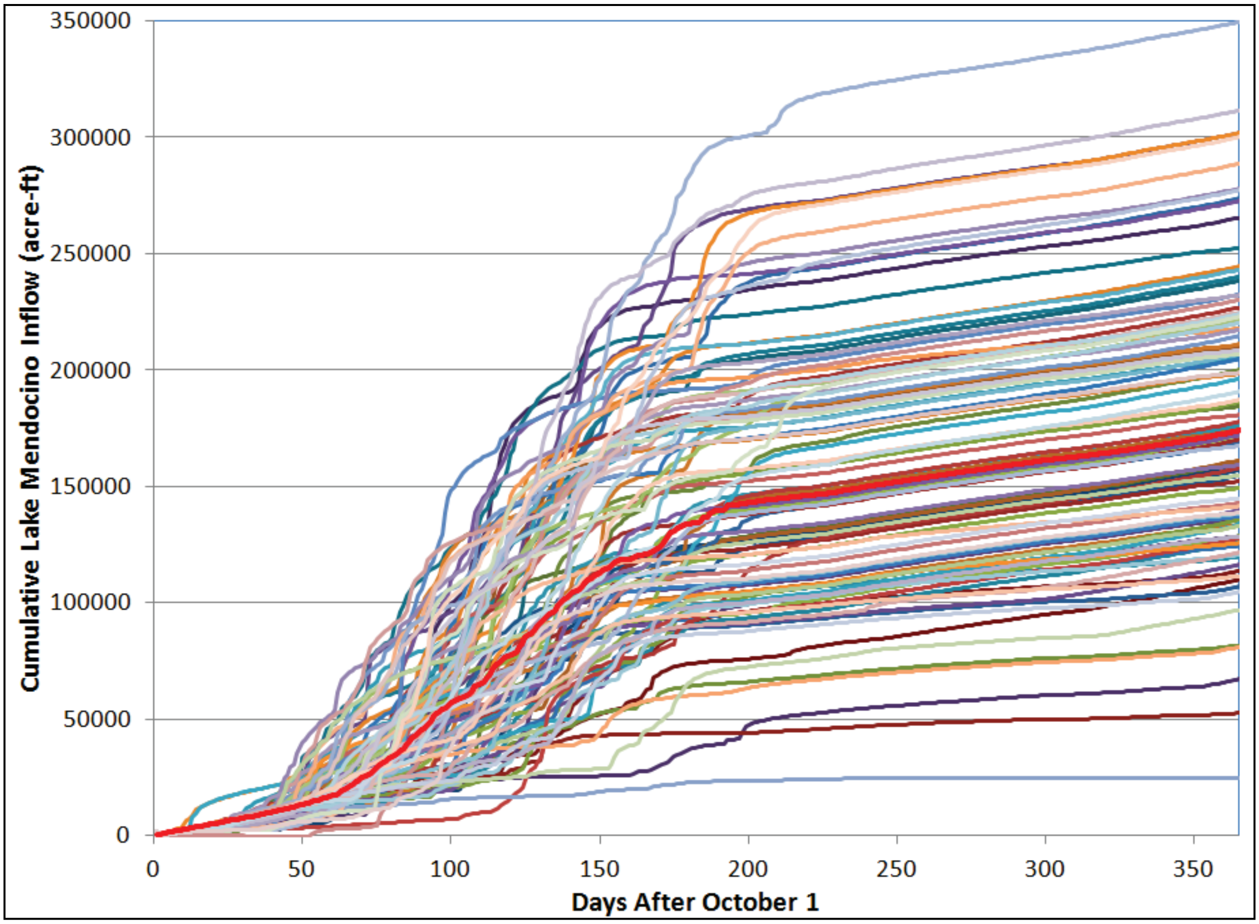

Figure 15. Cumulative Inflow into Lake Mendocino for 98-year period of Record.

only looking at cumulative inflows, not storage. Cumulative inflow thresholds are contained in Table 12.

Table 12. Lake Mendocino Cumulative Inflow Thresholds.

Cumulative Inflow into Lake Mendocino (Acre-Feet) as of						
	1/1	2/1	3/1	4/1	5/1	6/1
Normal	>22,500	>54,000	>72,000	>93,500	>99,000	>102,000
Dry	<22,500	<54,000	<72,000	<93,500	<99,000	<102,000
Critical	<19,250	<37,000	<49,000	<54,000	<54,000	<54,000

In addition, this analysis used storage in Lake Mendocino to modify the hydrologic index; if the storage was less than 69,000 acre-feet on May 31, then the index was changed from Normal to Dry, and if storage in Lake Mendocino dropped below 30,000 acre feet between October 1 and December 31, the index was changed from Normal to Dry. The 69,000 acre-feet storage threshold could be modified to get the desired percentage of Normal and Dry periods. Using a May 31 storage of 69,000 acre-feet resulted in a Dry index that was observed 34.5 percent of the time; this is much higher than was observed historically.

The logic for this index approach was added to the HEC-ResSim model as a state variable script (contained in Appendix A.1). During a simulation, the script computes the cumulative inflow into Lake Mendocino and compares it to the thresholds to determine the hydrologic index. HEC-ResSim uses the computed hydrologic index to set minimum stream flows. The computed hydrologic index was saved as part of the simulation. This time-series of hydrologic index (Normal, Dry, and Critical) for the 98-year simulation reflects the current hydrologic index applied directly to the East Fork of the Russian River.

Figure 16 shows a portion of the computed time-series of cumulative inflow into Lake Mendocino (using current PVP water); the new hydrologic index computed using Lake Mendocino inflow and the old hydrologic index using Lake Pillsbury inflow. The overall patterns are similar; however, there are differences. A hydrologic index tied to the Russian River watershed would do a better job responding to the hydrologic conditions within this watershed. Also, a cumulative inflow index will automatically respond to any future changes to PVP water.

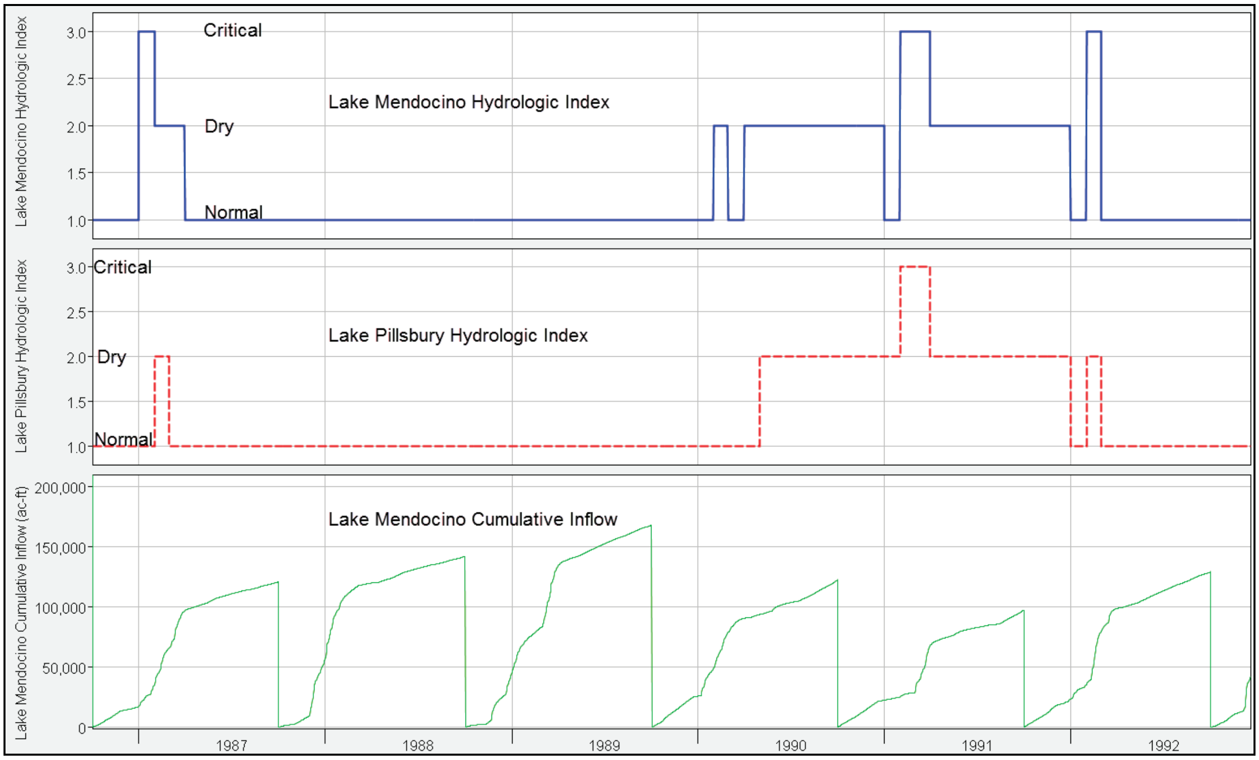

Figure 16. Comparison of Hydrologic Index results.

4.3 Lake Mendocino Storage Hydrologic Index

An alternative hydrologic index was developed by evaluating Lake Mendocino storage on the beginning of the month from January 1 through June 1. In addition, logic was included to change the index from Normal to Dry if storage in Lake Mendocino falls below 30,000 acre-feet between October 1 and December 31. This hydrologic index approach was added to the HEC-ResSim model as a state variable script (contained in Appendix A.2). During a simulation, HEC-ResSim compares the current reservoir storage to the storage thresholds to determine the hydrologic index. The hydrologic index is then used to set minimum stream flow requirements.

Figure 17 shows modeled reservoir storage at Lake Mendocino from the base simulation (98 years). This dataset was used initially to estimate storage thresholds for setting the hydrologic index. The storage thresholds were then modified by trial and error to create similar monthly percentages of Normal, Dry, and Critical periods as observed using the current hydrologic index. Table 13 contains the storage thresholds, and Figure 18 shows them plotted along with the guide curve. These storage thresholds result in a similar

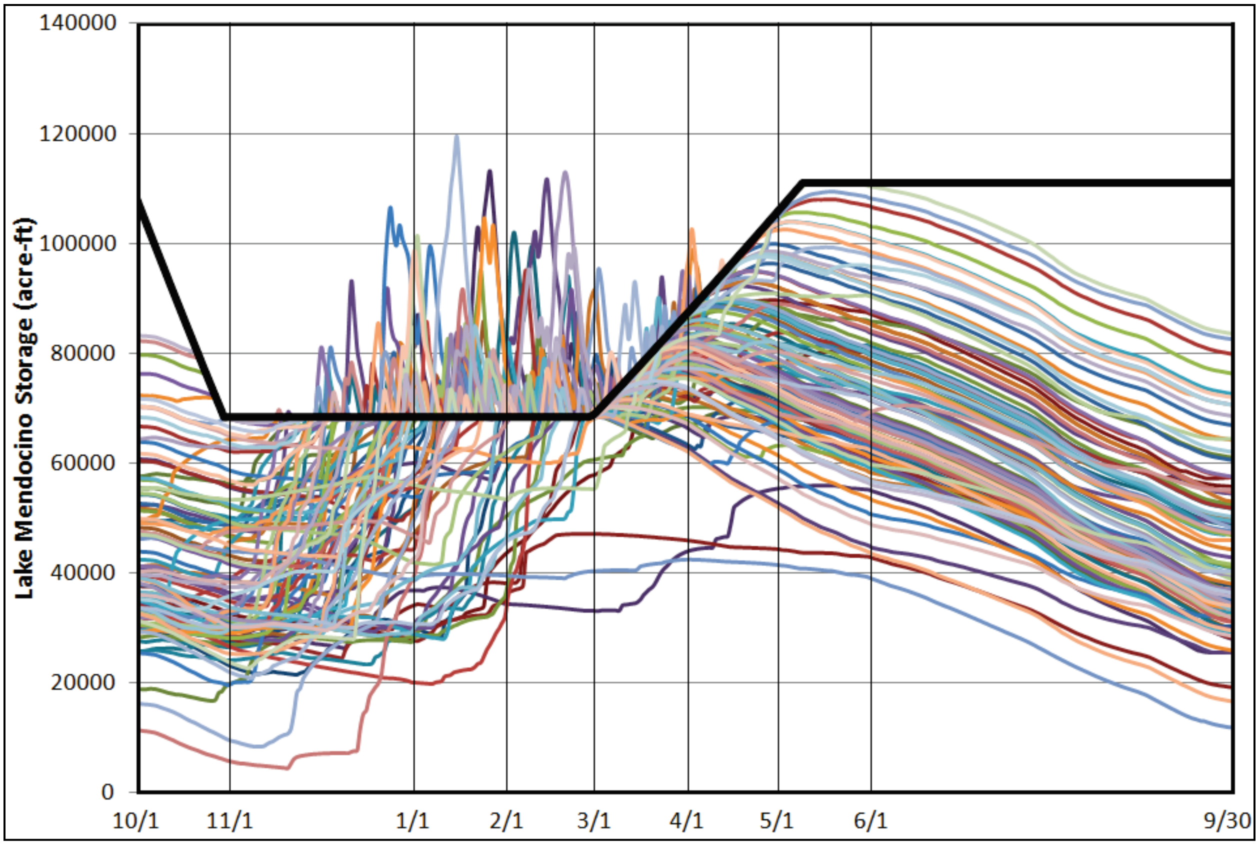

Figure 17. Lake Mendocino Storage from 98-year period of Record, Base Simulation.

Table 13. Lake Mendocino Storage Thresholds.

	Storage in Lake Mendocino (Acre-Feet) as of					
	1/1	**2/1**	**3/1**	**4/1**	**5/1**	**6/1**
Normal	>40,400	>67,700	>68,300	>71,700	>66,200	>61,000
Dry	<40,400	<67,700	<68,300	<71,700	<66,200	<61,000
Critical	<34,200	<52,700	<58,600	<56,00	<49,800	<44,400

percentage of Normal, Dry, and Critical periods as the base analysis. The storage thresholds could be modified to obtain different percentages of Normal, Dry, and Critical periods.

As Figure 17 shows, storage in Lake Mendocino typically reaches the guide curve (68,400 acre-feet) by March 1. This is due to fall/winter rains along with PVP water filling the reservoir. Also, storage generally peaks before May 1 and then decreases when Winter rains begin in the Fall/Winter. This explains why the storage thresholds increase from January 1 through April 1 and then decrease in May and June.

4.4 Lake Mendocino Cumulative Inflow Evaluated Weekly Hydrologic Index

An alternative hydrologic index was developed by modifying the cumulative Inflow Lake Mendocino index so that it was evaluated weekly (every Monday) from January 1 through

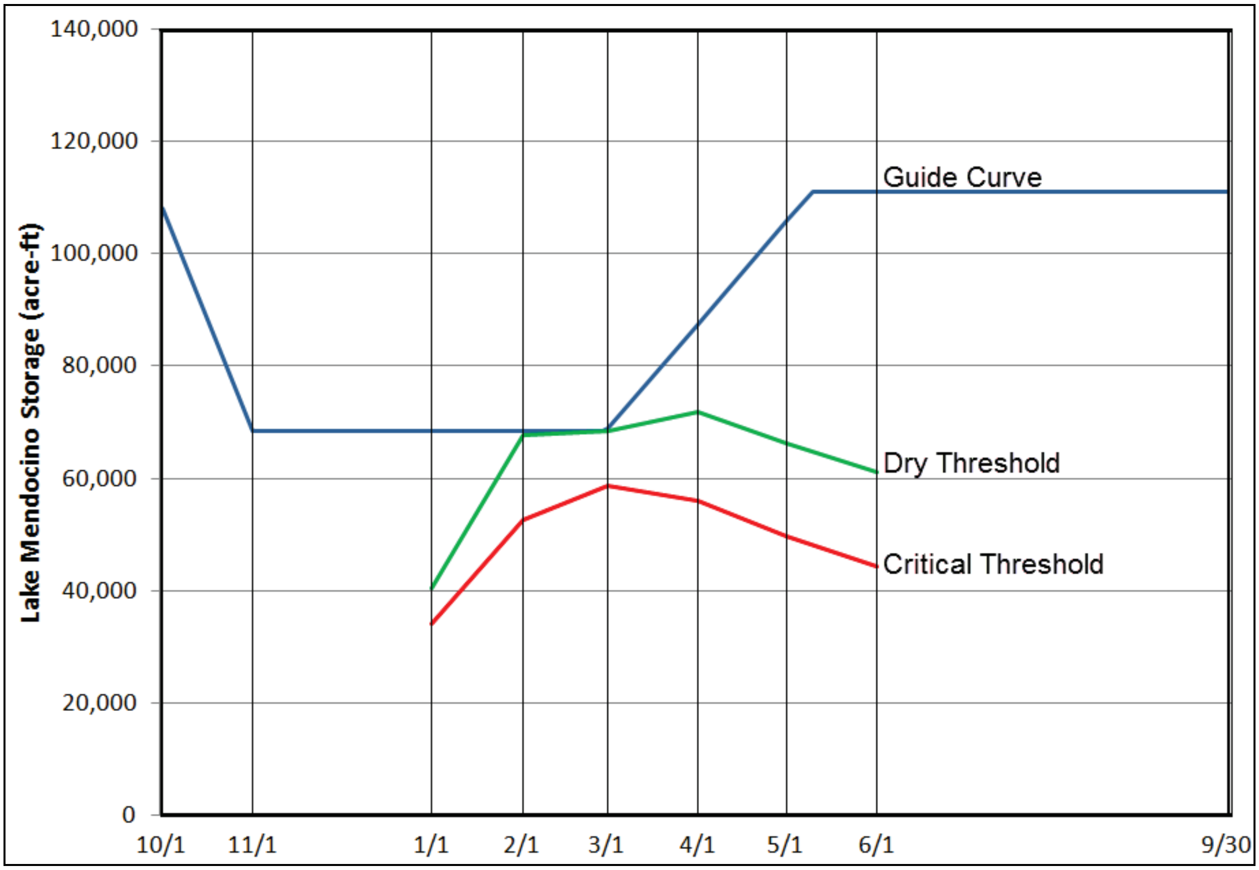

Figure 18. Lake Mendocino Storage Thresholds with Guide Curve.

June 1. The cumulative inflow into Lake Mendocino is compared to thresholds to determine the hydrologic index. In addition, this index evaluates the storage in Lake Mendocino to modify the index; if the storage is less than a threshold on May 31, the index will change from Normal to Dry, and if storage in Lake Mendocino drops below 30,000 acre-feet between October 1 and December 31, the index will change from Normal to Dry. The cumulative inflow thresholds and the May 31 storage threshold can be modified to obtain the required percentages of Normal, Dry, and Critical periods from the 98-year period-of-record dataset.

The monthly percentages of Normal, Dry, and Critical from the current hydrologic index (cumulative inflow into Lake Pillsbury) were used to set thresholds for cumulative inflow into Lake Mendocino. Instead of thresholds on the first of the month, thresholds were developed using annual patterns based on percentiles from the cumulative inflow dataset. This inflow dataset was computed using the USGS unimpaired inflow data, simulated PVP water, and losses. Figure 19 shows the cumulative inflow into Lake Mendocino during the 98 period-of-record. The dark red lines show the third and fourteenth percentiles used for the Critical and Dry thresholds. The logic for this index approach was added to the HEC-ResSim model as a state variable script (contained in Appendix A.3). During a simulation, the script computes the cumulative inflow into Lake Mendocino and compares it to the annual pattern of cumulative inflow to compute the hydrologic index. The hydrologic index is then used to set minimum stream flows.

Figure 20 shows the hydrologic index during 1947. Notice how the index switches from Normal to Dry for one week durations in both February and April. SCWA had some concerns that weekly changes to the hydrologic index would adversely affect the biology within the

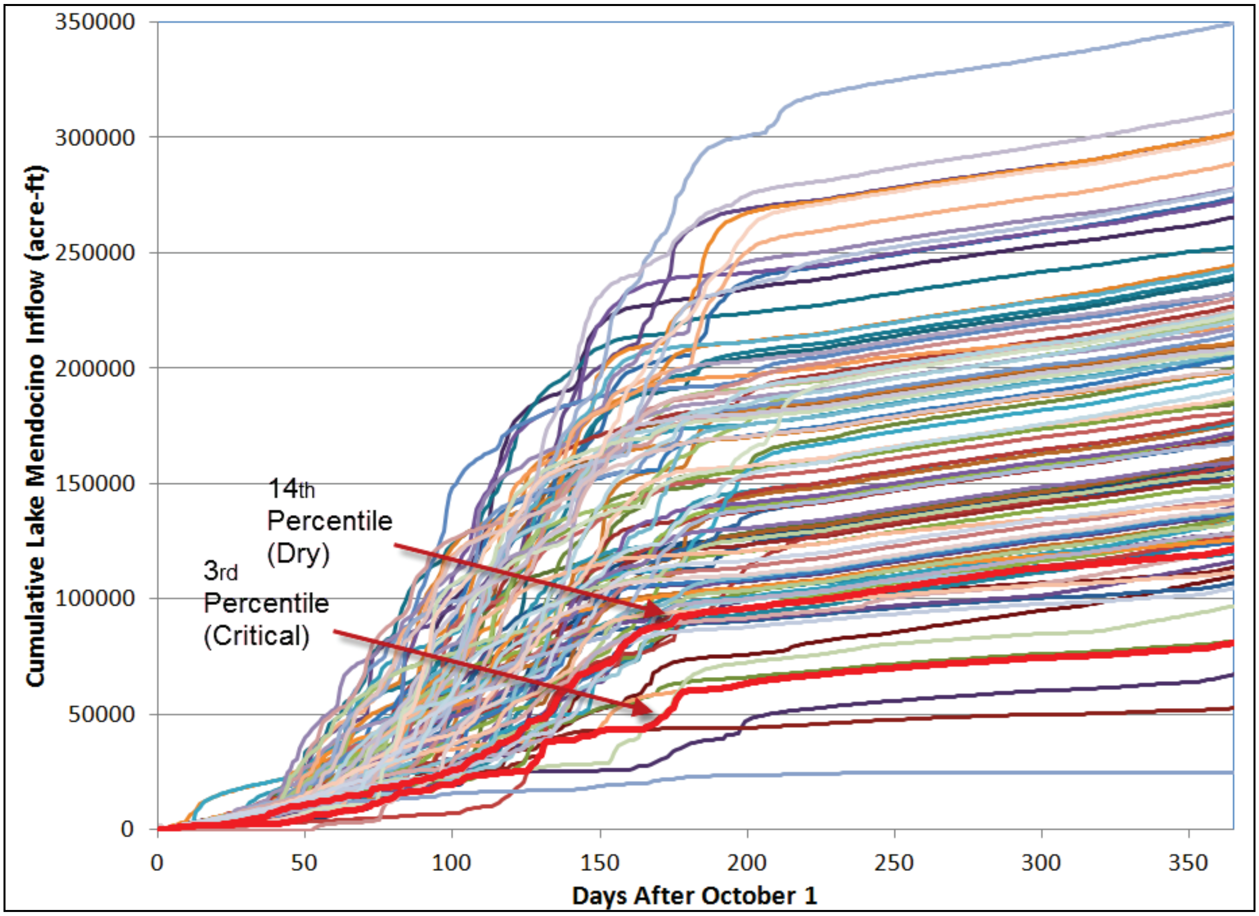

Figure 19. Lake Mendocino Cumulative Inflow Thresholds, Annual Patterns.

river, and it would be difficult to operate. Figure 21 shows a comparison during 1955 of storage in Lake Mendocino from model simulations where the hydrologic index was evaluated once a month and once a week (both are using cumulative inflow thresholds). For this year, the weekly evaluation generated a Dry hydrologic index during two weeks in April. This switch to a Dry index for fourteen days resulted in approximately 2,500 acre-feet of storage that was kept within Lake Mendocino for the 1955 water year.

4.5 Lake Mendocino Storage Evaluated Weekly Hydrologic Index

An alternative hydrologic index was developed by modifying the storage Lake Mendocino hydrologic index so that it was evaluated weekly (every Monday) for the entire water year. Lake Mendocino storage is compared to thresholds to determine the hydrologic index. The storage thresholds can be modified to obtain the required percentages of Normal, Dry, and Critical periods from the 98-year period-of-record dataset.

Instead of thresholds on the first of the month, thresholds were developed using annual patterns based on percentiles of Lake Mendocino Storage from the base dataset. Figure 22 shows the storage in Lake Mendocino during the base simulation 98-year period-of-record. The dark red lines show the third and fourteenth percentiles used for the Critical and Dry thresholds. The logic for this index approach was added to the HEC-ResSim model as a state variable script contained in Appendix A.4). During a simulation, the script compares the current storage in Lake Mendocino to the annual storage patterns to compute the hydrologic index. The hydrologic index is then used to set minimum stream flows.

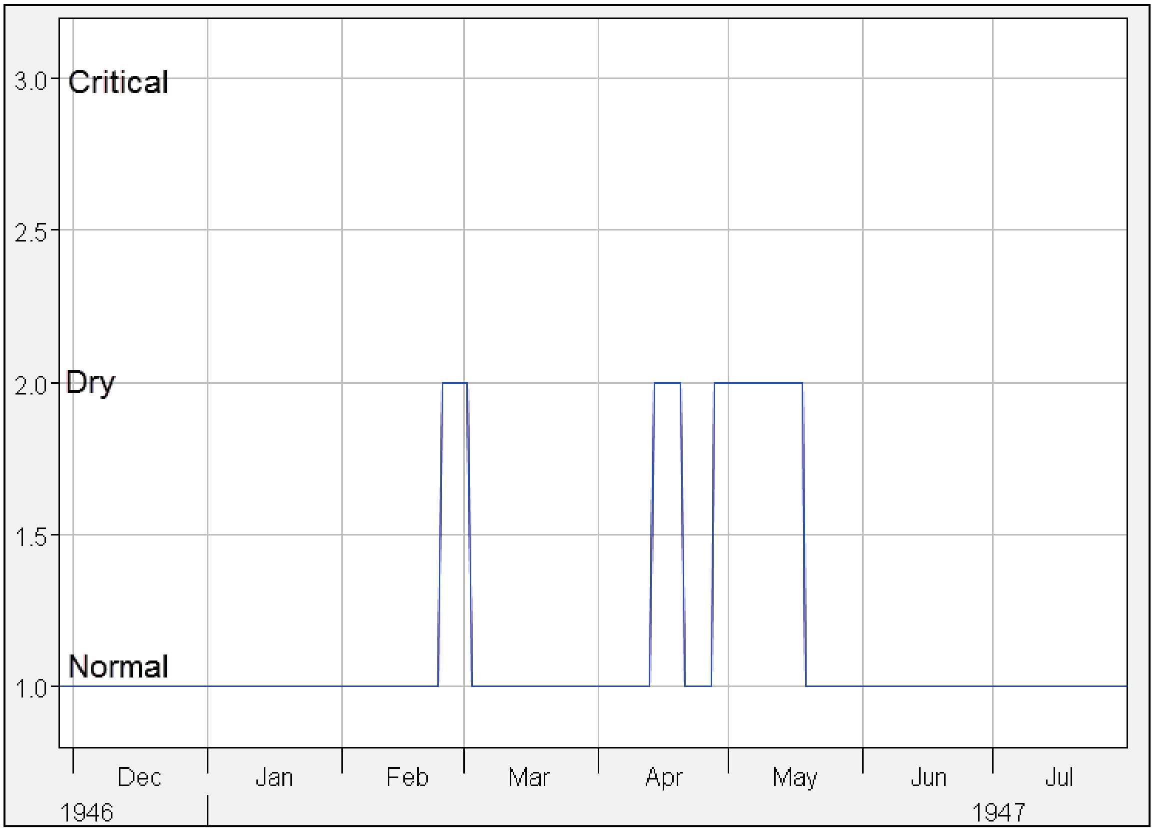

Figure 20. Hydrologic Index when Cumulative Inflow was Evaluated Weekly.

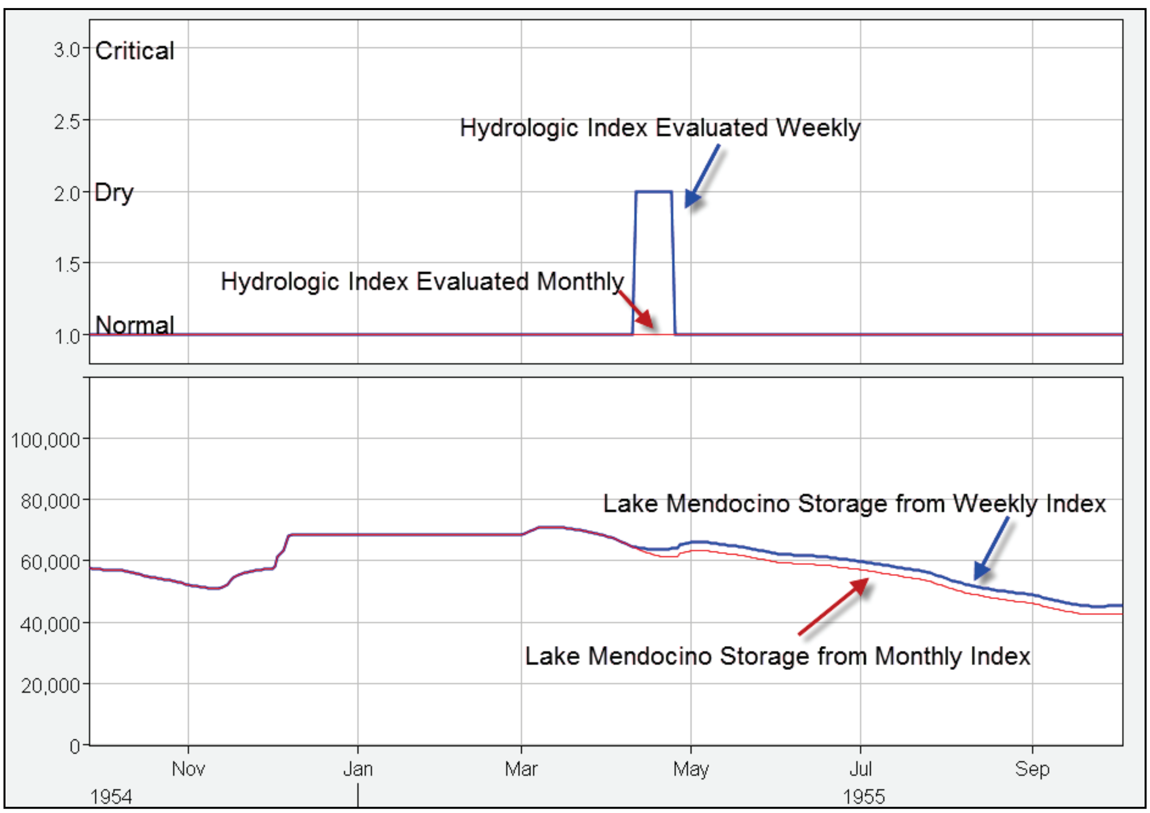

Figure 21. Comparison of Monthly and Weekly Cumulative Inflow Index.

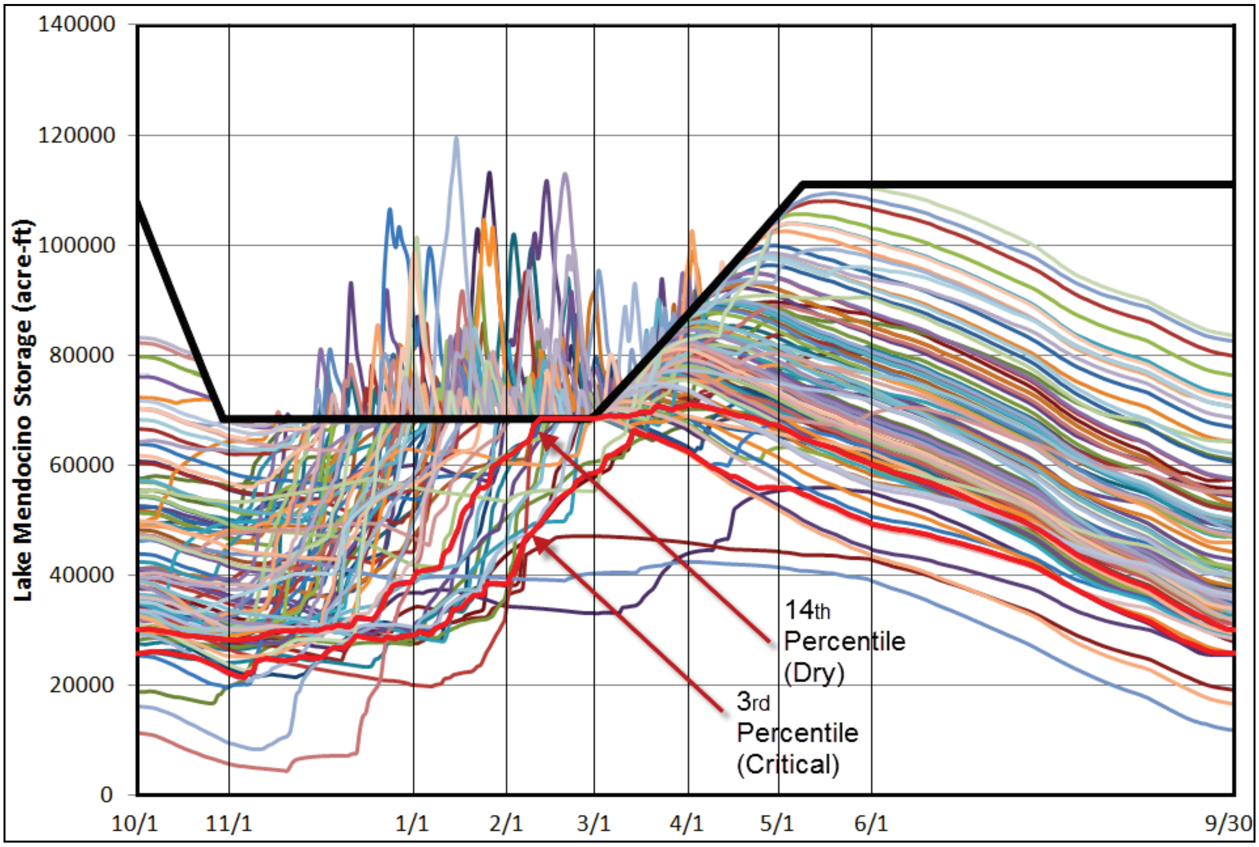

Figure 22. Lake Mendocino Storage Thresholds, Annual Pattern.

Figure 23 shows the hydrologic index during 1946. Notice how the index switches from Normal to Dry six times between April and August. As with the weekly cumulative inflow hydrologic index, SCWA had some concerns that frequent changes to the hydrologic index would adversely affect the biology within the river, and it would be difficult to operate. Figure 24 shows a comparison during 1964 of the hydrologic index from model simulations where the index was evaluated once a month and once a week (both are using Lake Mendocino storage thresholds). Overall, the percent of Normal and Dry periods are similar; the difference is when in time the hydrologic index switches from Normal to Dry.

4.6 Dual Index – Cumulative Lake Mendocino Inflow and Lake Sonoma Storage Hydrologic Index

An alternative hydrologic index was developed by creating separate hydrologic index criteria for Lake Sonoma and Lake Mendocino. Releases from Lake Mendocino are used to meet minimum stream flow requirements from the dam to the confluence of the Russian River and Dry Creek (Upper Russian River). Releases from Lake Sonoma are used to meet minimum stream flow requirements in Dry Creek and from the confluence of Dry Creek and the Russian River to the ocean (Lower Russian River). In effect, the upper river could be in a Normal hydrologic condition, while the lower river could be in a Critical hydrologic condition.

For the Upper Russian River, cumulative inflow into Lake Mendocino, along with storage in Lake Mendocino on May 31 and October 1 through December 31, were used to determine the hydrologic index (the same HEC-ResSim script developed for monthly evaluation of cumulative inflow into Lake Mendocino was applied, Appendix A.1). The cumulative inflow was compared to thresholds, contained in Table 12, to determine the hydrologic index from

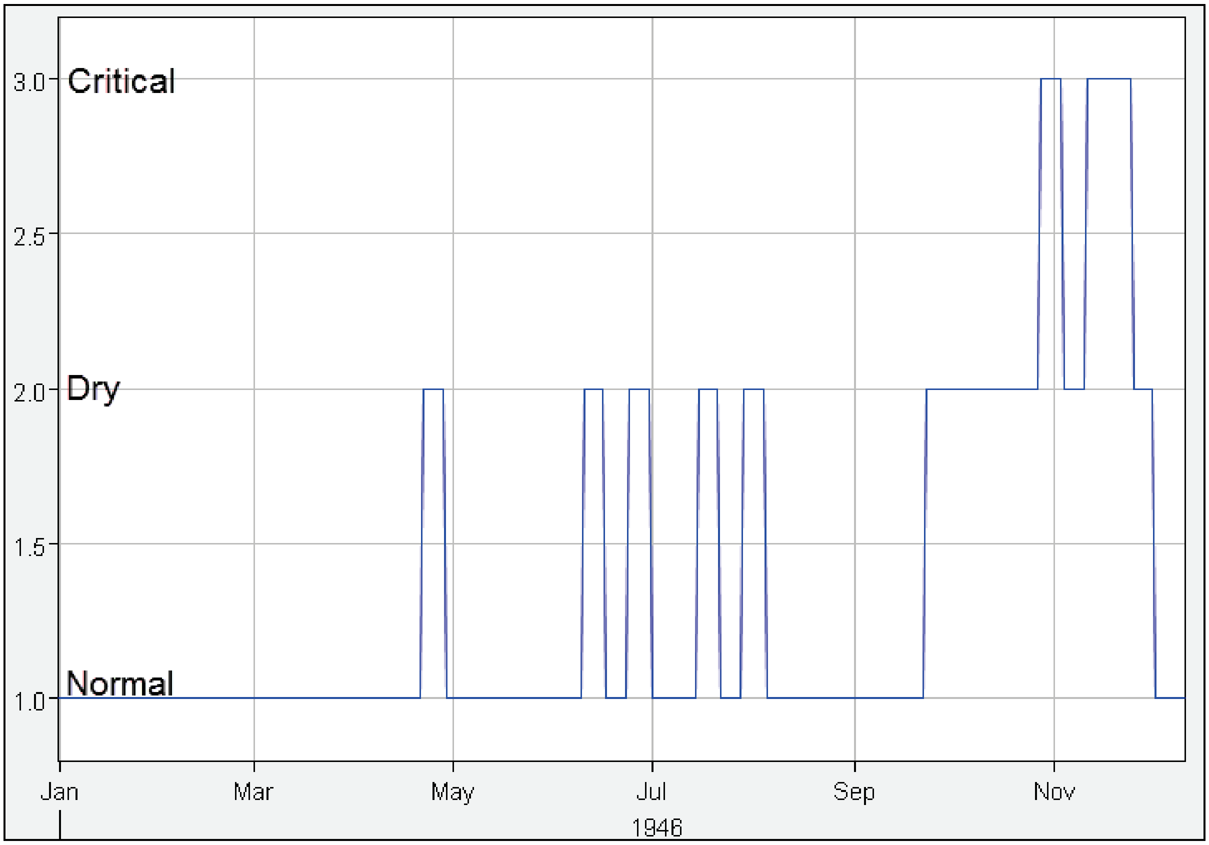

Figure 23. Hydrologic Index when Storage was Evaluated Weekly.

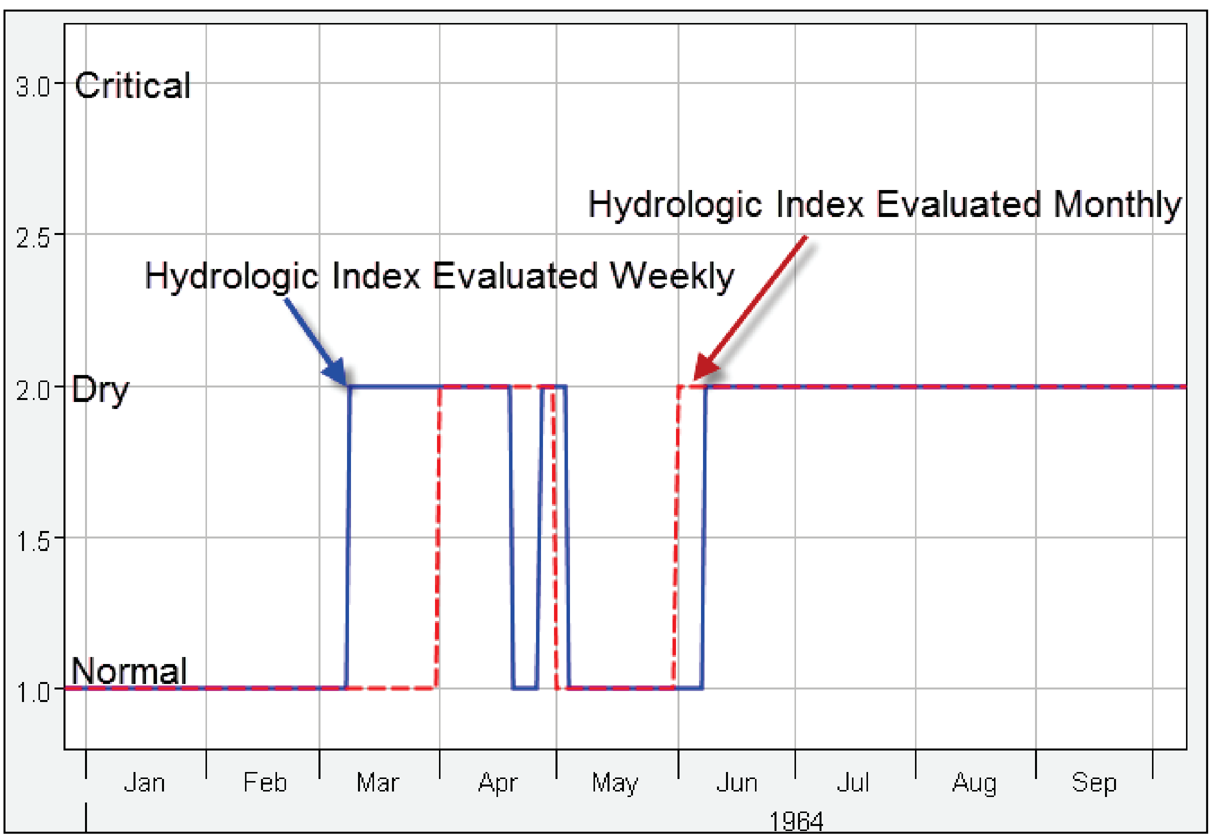

Figure 24. Comparison of Monthly and Weekly Lake Mendocino Storage Hydrologic Index.

January 1 to June 1 (evaluated monthly). If the storage on May 31 was less than a threshold, a Normal index could be changed to Dry. Also, if storage in Lake Mendocino fell below 30,000 acre-feet, a Normal index could be changed to Dry.

Instead of cumulative inflow, a separate criterion was used for determining a hydrologic index for the Lower Russian River. Lake Sonoma is able to carry over a significant amount of water from one year to the next because it contains a large amount of storage when compared to demand. Because of this, it seems more reasonable to use storage when setting the hydrologic index for the lower river. Model simulations show that the reservoir will have a large amount of water stored (enough to meet demands, even though rainfall-runoff was critically low for one or two years). Therefore, storage in Lake Sonoma was evaluated weekly for the entire water year to determine the hydrologic index in the lower river. The storage thresholds used to determine the hydrologic index could be modified to obtain the required percentages of Normal, Dry, and Critical periods from the 98-year period-of-record dataset.

Storage thresholds were developed using annual patterns based on percentiles of Lake Sonoma Storage from the base dataset. Figure 25 shows the storage in Lake Sonoma during the base simulation 98-year period-of-record. The dark red lines show the third and fourteenth percentiles used for the Critical and Dry thresholds. The logic for this index approach was added to the HEC-ResSim model as a state variable script (contained in Appendix A.5). During a simulation, the script compares the current storage in Lake Sonoma to the annual storage patterns to compute the hydrologic index. The hydrologic index is then used to set minimum stream flows in the lower river.

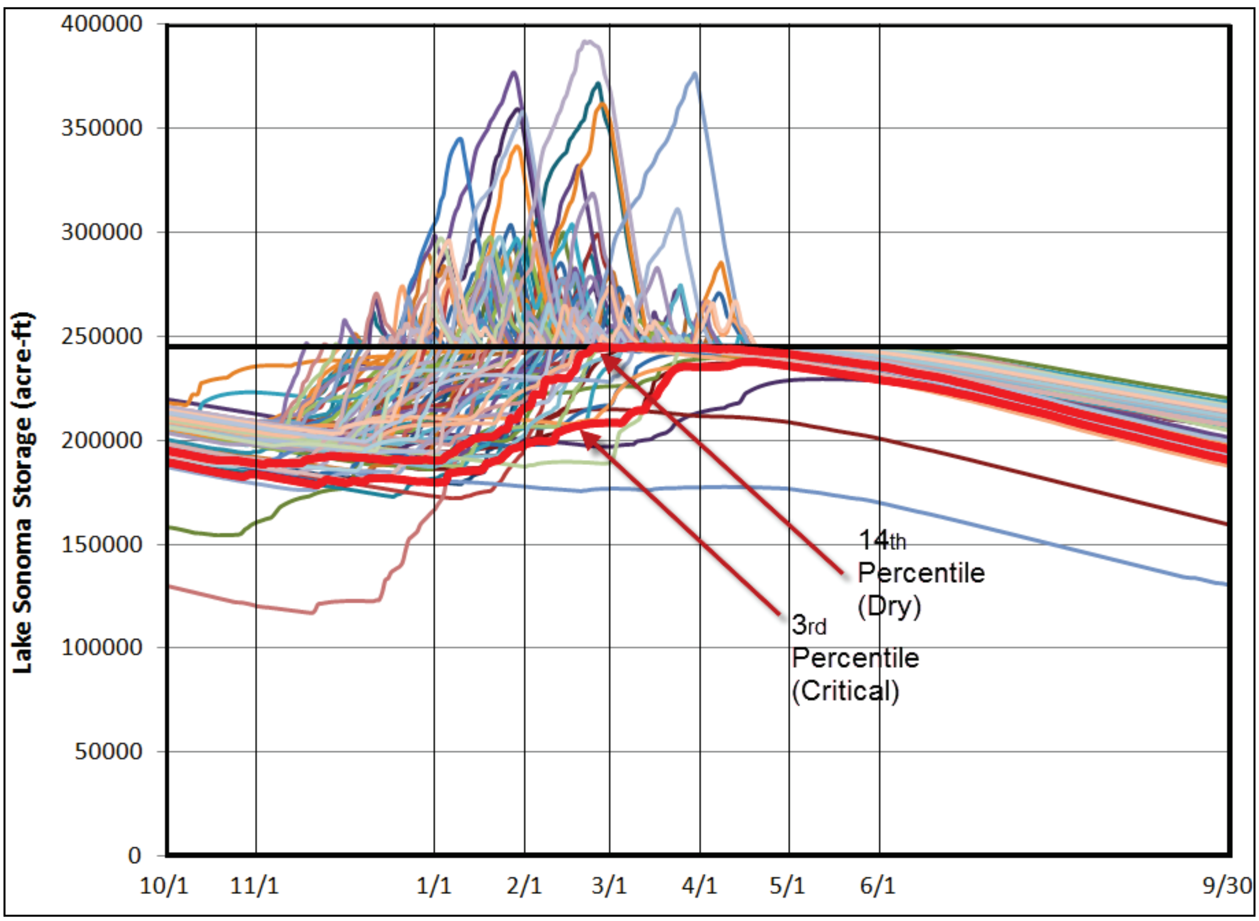

Figure 25. Lake Sonoma Storage Thresholds, Annual Pattern.

Figure 26 shows the hydrologic index for the upper river, using cumulative inflow and storage in Lake Mendocino, and the lower river, using storage in Lake Sonoma. Notice the hydrologic index in the upper river can be Dry while in the lower river it can be Normal. This results in different minimum stream flows in the Upper and Lower Russian River.

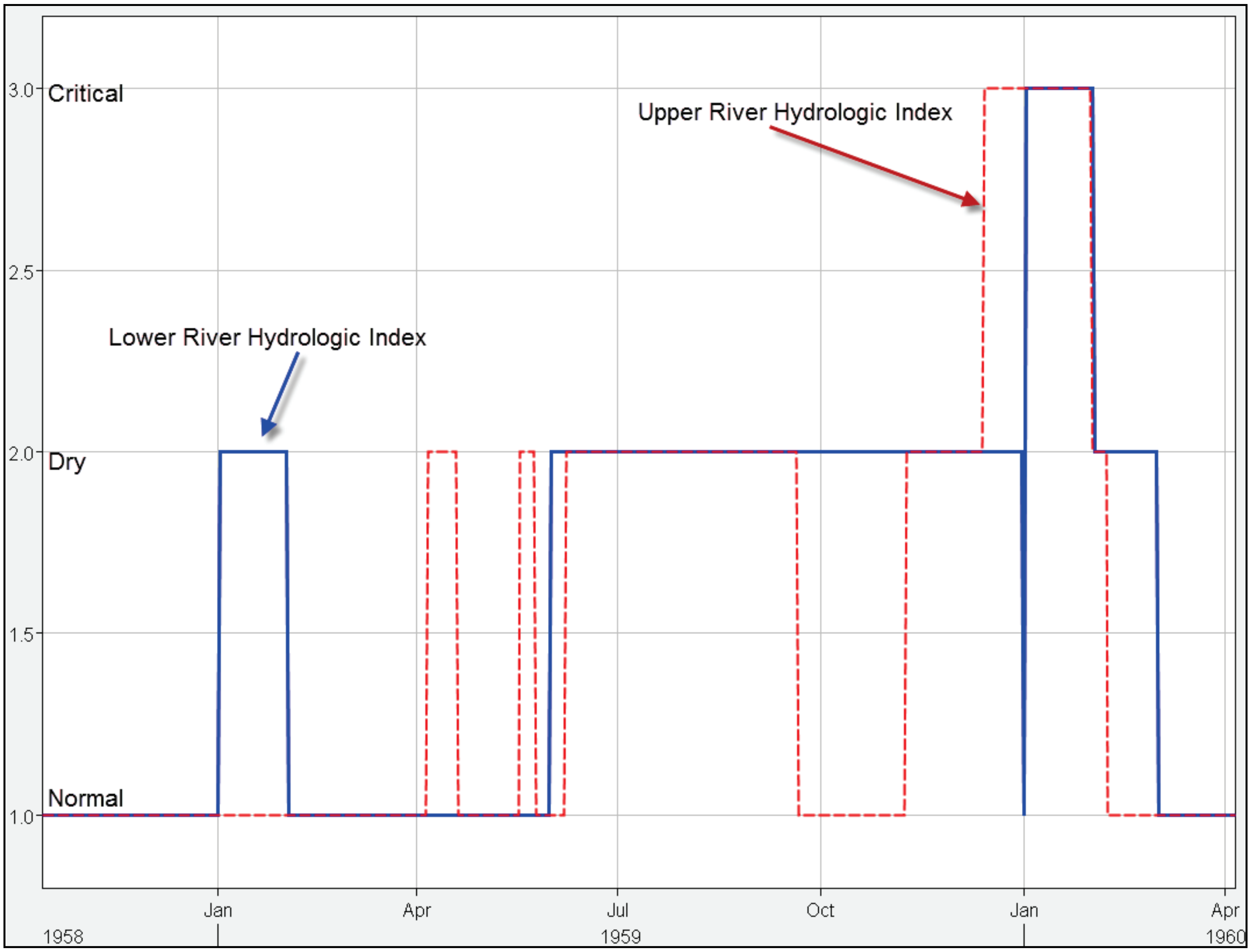

Figure 26. Dual Hydrologic Index Approach Results in Different Normal, Dry, and Critical Conditions for the Upper and Lower Russian River.

4.7 Water Balance Hydrologic Index

This hydrologic index alternative uses a water balance to determine the total amount of water available for meeting minimum flows, from the current time-step through the end of the water year. Water accounted for includes current reservoir storage, reservoir inflow, losses (including agricultural and SCWA losses plus lake evaporation) and minimum flow requirements. The analysis starts with the current month and then computes the monthly usable water (using monthly inflow, losses, minimum flow requirements, and storable water). The monthly usable water is summed for the year and then compared to the cumulative minimum flow requirement. The hydrologic index is selected so that the total usable water is able to meet the minimum flow requirements and downstream river losses.

For this analysis, the "loss" time-series developed by SCWA, at Redwood Valley, Hopland, Cloverdale, and Healdsburg, were used to compute the total monthly losses in the Upper Russian River. Figure 27 shows a schematic of the Russian River, and Figure 28 shows the loss time-series at each location. The loss time-series is treated as an annual pattern in the HEC-ResSim model. As previously stated, the loss time- series were the same for each

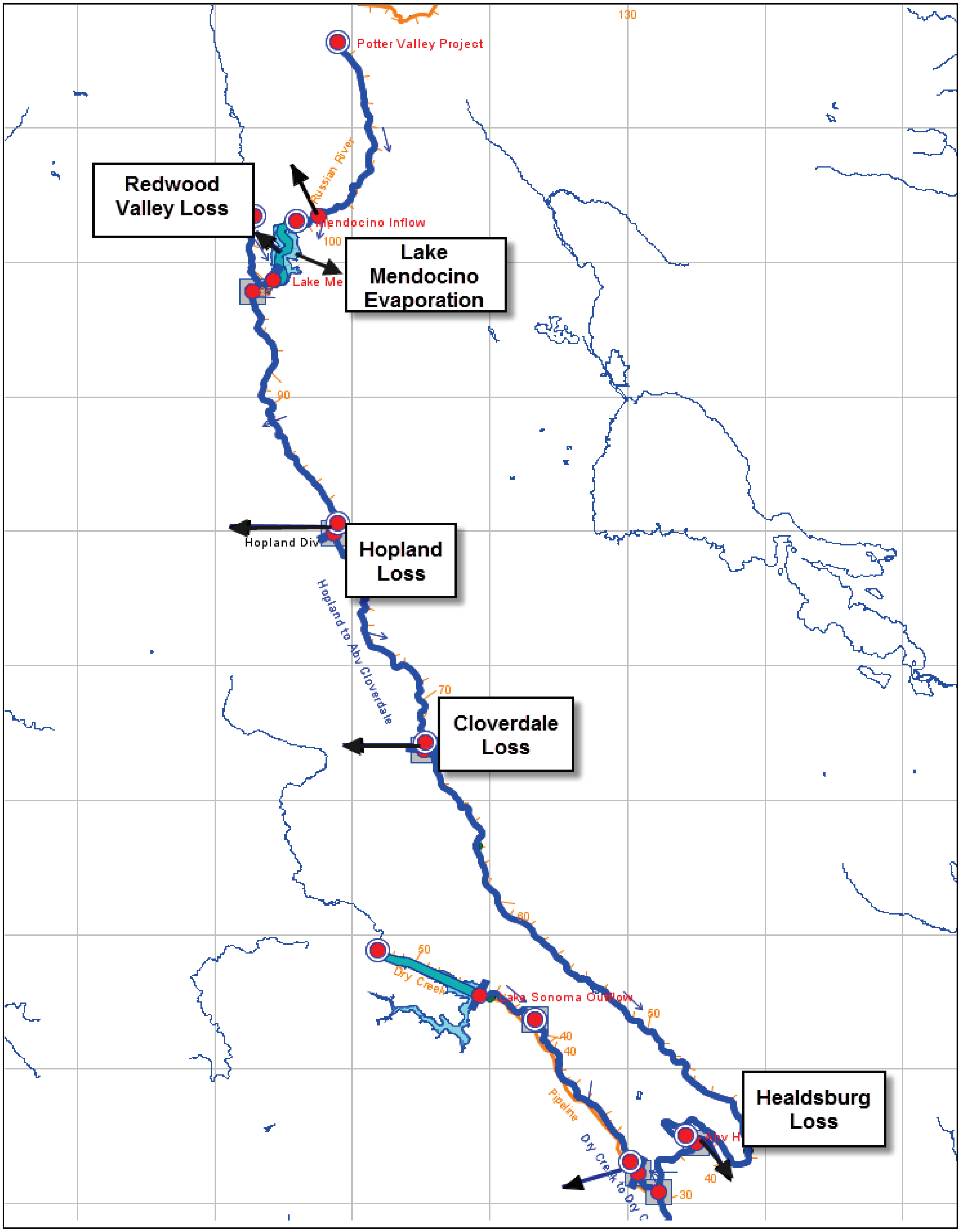

Figure 27. Losses in the Upper Russian River.

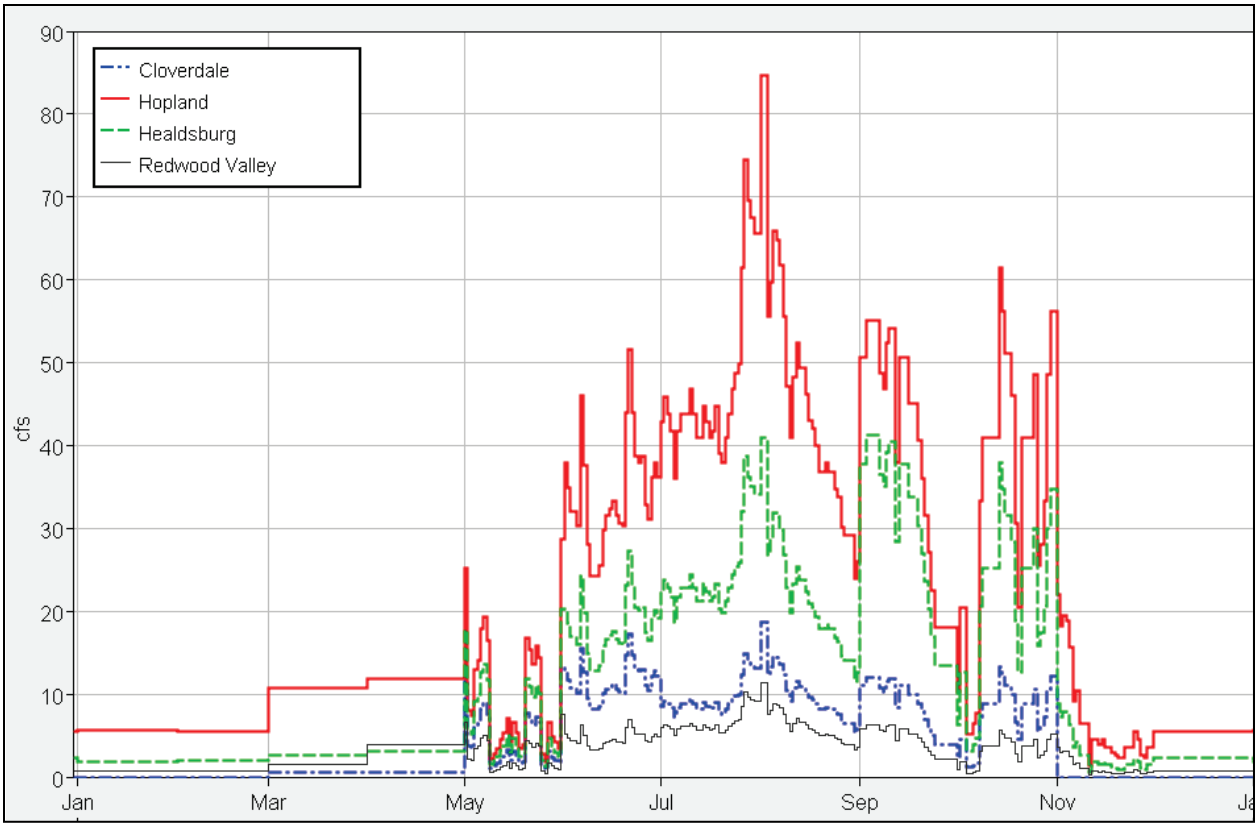

Figure 28. Daily Time-Series of Losses in the Upper Russian River.

water year (loss time-series are a multi-year average of losses estimated from an analysis of flow data plus known water supply). These loss time-series were converted to monthly volumes (acre-feet) and estimates of monthly evaporation from Lake Mendocino were used to compute the total monthly loss in the Upper Russian River. Table 14 contains the total monthly losses in the Upper Russian River watershed.

Table 14. Total Monthly Losses in the Upper Russian River.

Month	Losses plus Evaporation (acre-feet)
Jan	318
Feb	569
Mar	1,147
Apr	1,482
May	1,940
Jun	4,790
Jul	6,350
Aug	5,971
Sep	5,658
Oct	4,658
Nov	841
Dec	647

Using the current minimum flows for Normal, Dry, and Critical conditions, monthly volumes were computed (Table 15). For example, a volume of 11,375 acre-feet is required from Lake Mendocino to meet minimum flow requirements when the hydrologic index is Normal

Table 15. Monthly Minimum Flows and Volumes for Normal, Dry, and Critical Index Periods.

Month	Normal Minimum Flow (cfs)	Dry Minimum Flow (cfs)	Critical Minimum Flow (cfs)	Normal Minimum Volume (acre-feet)	Dry Minimum Volume (acre-feet)	Critical Minimum Volume (acre-feet)
Jan	150	75	25	9,223	4,612	1,537
Feb	150	75	25	8,330	4,165	1,388
Mar	150	75	25	9,224	4,612	1,538
Apr	185	75	25	11,008	4,463	1,487
May	185	75	25	11,375	4,611	1,537
Jun	125	75	25	7,438	4,463	1,488
Jul	125	75	25	7,686	4,612	1,537
Aug	125	75	25	7,686	4,611	1,537
Sep	125	75	25	7,438	4,463	1,488
Oct	125	75	25	7,438	4,463	1,488
Nov	150	75	25	7,438	4,463	1,488
Dec	150	75	25	7,438	4,463	1,488

during the month of May. This is reduced to 4,611 acre-feet when the index is Dry and increased to 1,537 acre-feet when the index is Critical.

Monthly inflow into Lake Mendocino was also included in the water balance. Monthly inflows could be generated from flow forecasts, historic data, or models. For this study, the cumulative inflow into Lake Mendocino dataset, shown in Figure 15, (USGS estimated unimpaired inflows plus PVP diversions minus losses in the basin upstream of Lake Mendocino from 1910 – 2008) was used to compute multiple percentiles (tenth, twenty-fifth, and fiftieth) of inflow for each month, contained in Table 16. Initially, the fiftieth percentile inflow was used in the water balance; however, options that incorporate hydrologic or meteorologic conditions were included.

Table 16. Percentile Inflows into Lake Mendocino.

	Inflow 10^{th} Percentile acre-feet	Inflow 25^{th} Percentile acre-feet	Inflow 50^{th} Percentile acre-feet
Jan	12,508	20,111	35,912
Feb	11,372	20,403	29,854
Mar	5,447	11,129	17,321
Apr	2,937	3,441	4,714
May	3,707	5,500	5,561
Jun	2,824	5,777	5,785
Jul	2,372	5,445	5,445
Aug	2,421	5,465	5,465
Sep	4,255	4,485	6,537
Oct	2,508	7,535	7,921
Nov	5,633	8,311	9,871
Dec	5,288	12,736	27,191

Initially, a spreadsheet was created to illustrate the water balance approach. Figure 29 shows the input data in the spreadsheet, and Figure 30 shows an example calculation for the water balance approach. The input data includes inflow, losses, end of month (EOM) guide curve, and incremental and cumulative minimum flows. The following describes each column in Figure 30:

Inflow, Losses, and End of Month Guide Curve (Conservation Storage)						
		Forecasted Inflows				EOM
		Inflow 10%	Inflow 25%	Inflow 50%	Loss + Evap	Guide Curve
	Month	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft
1	Jan	12508	20111	35912	618	68409
2	Feb	11372	20403	29854	569	68409
3	Mar	5447	11129	17321	1147	86400
4	Apr	2937	3441	4714	1482	105930
5	May	3707	5500	5561	1940	110967
6	Jun	2824	5777	5785	4790	110967
7	Jul	2372	5445	5445	6350	110967
8	Aug	2421	5465	5465	5971	110967
9	Sep	4255	4485	6537	5658	110967
10	Oct					68409
11	Nov					68409
12	Dec					68409

Volumes Needed for Min Flow for Each Condition						
	Monthly			Cumulative		
	Normal Min	Dry Min	Critical Min	Normal Min	Dry Min	Critical Min
	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft
Jan	9223	4612	1537	79408	40612	13537
Feb	8330	4165	1388	70185	36000	12000
Mar	9224	4612	1538	61855	31835	10612
Apr	11008	4463	1487	52631	27223	9074
May	11375	4611	1537	41623	22760	7587
Jun	7438	4463	1488	30248	18149	6050
Jul	7686	4612	1537	22810	13686	4562
Aug	7686	4611	1537	15124	9074	3025
Sep	7438	4463	1488	7438	4463	1488
Oct						
Nov						
Dec						

Figure 29. Input Data for Water Balance Spreadsheet.

1) Column 1 is the Month. The analysis starts in the current month (assume the beginning of the month) and computes the usable water for each month until the end of the water year. The example shown in Figure 30 begins the water balance in January.
2) Column 2 is the Monthly Inflow. This is the fiftieth, twenty-fifth, or tenth percentiles. Without a forecast, the fiftieth percentile is selected. The example shown in Figure 30 uses the fiftieth percentile inflow (copied from the input data table).
3) Column 3 is monthly downstream loss and lake evaporation (the total loss from the Upper Russian River). The data in this column is copied from the input table.
4) Column 4 is the minimum flow requirements. These values can change based on the forecast, like inflow. The example shown in Figure 30 uses the Normal condition minimum flows (copied from the input data table).

			Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
Month	1				Loss and		End of Month		
Beginning	30000			Monthly Inflow	Evap	Minimum Flow	Storage	Water Stored	Monthly Usable
Forcecast	1		Month	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft
		1	Jan	35,912	618	9,223	56,071	26,071	35,912
		2	Feb	29,854	569	8,330	68,409	12,338	21,237
		3	Mar	17,321	1,147	9,224	75,359	6,950	17,321
		4	Apr	4,714	1,482	11,008	67,583	0	4,714
		5	May	5,561	1,940	11,375	59,829	0	5,561
		6	Jun	5,785	4,790	7,438	53,385	0	5,785
		7	Jul	5,445	6,350	7,686	44,795	0	5,445
		8	Aug	5,465	5,971	7,686	36,603	0	5,465
		9	Sep	6,537	5,658	7,438	30,044	0	6,537
		10	Oct						
		11	Nov						
		12	Dec						
								Sum	107,977
								Buffer	0
								Total Available Water	137,977
								Total Loss	28,525
								Water Available for Minimum Flow	109,452
								INDEX	1

Figure 30. Example Water Balance Spreadsheet Starting in January with Initial Lake Mendocino Storage of 30,000 acre-feet.

5) Column 5 is the End of Month Storage. This is computed by taking the minimum of the end of month guide curve (conservation storage) or the Previous Month Storage plus Monthly Inflow (Column 2) – Loss (Column 3) – Minimum Flow (Column 4). The water balance cannot use (store) more water than is available at the conservation storage.
6) Column 6 is the Water Stored. This is computed by subtracting the End of Month Storage from the beginning month storage.
7) Column 7 is the Monthly Usable water. This is the minimum of the Monthly Inflow (Column 2) or Loss (Column 3) plus Minimum Flow (Column 4) plus Water Stored (Column 6). This is the water that can be used to meet minimum flows and losses, and it accounts for the fact that Lake Mendocino cannot store water when the reservoir is at guide curve (conservation storage).

The usable monthly water is summed from the beginning of the analysis to the end of the water year. In the example shown in Figure 30, the total usable water is 107,977 acre-feet (January through September). The Total Available Water is the total usable water plus the storage in Lake Mendocino at the beginning of the simulation (in this example, 30,000 acre-feet). Total Loss is the cumulative loss plus evaporation from the beginning of the simulation until the end of the year. The Water Available for Minimum Flow is the Total Available Water subtracting Total Loss. The hydrologic index is evaluated by comparing the computed Water Available for Minimum Flow to the cumulative minimum flows (highlighted in green, orange, and red in Figure 29). For example, 79,408 acre-feet are required to meet minimum flows in the Upper Russian River for an analysis period from January through September when the hydrologic index is Normal. In the example shown in Figure 30, the amount of water that is available for meeting minimum flows, 109,452 acre-feet, is greater than 79,408 acre-feet; therefore, the reservoir can meet minimum flows assuming a Normal hydrologic index. If the amount of water available for meeting minimum flows was less than 79,408 acre-feet and greater than 40,612 acre-feet, then a Dry index would be selected. If the amount of water available for meeting minimum flows was less than 40,612 acre-feet, then a Critical index would be selected.

The example water balance shown in Figure 31 is similar to the one shown in Figure 30. The example begins in January; however, a Critical forecast was specified. Notice Column 2 is much different; the tenth percentile monthly inflows were used instead of the fiftieth percentile. In this example, the amount of water available for meeting minimum flows is 49,319 acre-feet (the result is a Dry index).

				Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
Month	1					Loss and		End of Month		
Beginning	30000				Monthly Inflow	Evap	Minimum Flow	Storage	Water Stored	Monthly Usable
Forcecast	3			Month	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft
			1	Jan	12,508	618	1,537	40,353	10,353	12,508
			2	Feb	11,372	569	1,388	49,768	9,415	11,372
			3	Mar	5,447	1,147	1,538	52,530	2,762	5,447
			4	Apr	2,937	1,482	1,487	52,498	0	2,937
			5	May	3,707	1,940	1,537	52,727	230	3,707
			6	Jun	2,824	4,790	1,488	49,274	0	2,824
			7	Jul	2,372	6,350	1,537	43,759	0	2,372
			8	Aug	2,421	5,971	1,537	38,672	0	2,421
			9	Sep	4,255	5,658	1,488	35,782	0	4,255
			10	Oct						
			11	Nov						
			12	Dec						
									Sum	47,844
									Buffer	0
									Total Available Water	77,844
									Total Loss	28,525
									Water Available for Minimum Flow	49,319
									INDEX	2

Figure 31. Example Water Balance Spreadsheet Starting in January with Initial Lake Mendocino Storage of 30,000 acre-feet and a "Critical" Forecast.

The example water balance shown in Figure 32 begins in April. Notice that there are no calculations prior to April in the table. A buffer option is used in this example. The buffer is an attempt to ensure the reservoir does not empty below a certain threshold. The buffer can also be thought of as a "tuning knob" that can be used to control the number of Normal, Dry, and Critical periods. The buffer is applied when computing the total available water. When using the buffer, the total available water is the total monthly usable water plus the storage in Lake Mendocino at the beginning of the simulation minus the buffer amount.

This water balance hydrologic index was added to the HEC-ResSim model as a state variable script (contained in Appendix A.6). During a simulation, HEC-ResSim computes the water available to meet minimum flows to determine the hydrologic index (following a similar procedure as shown in the spreadsheet). The hydrologic index was used by HEC-ResSim to set minimum stream flows throughout the Russian River model. Options, tuning knobs, were added to the water balance approach within the HEC-ResSim model to provide flexibility in how many Normal, Dry, and Critical periods were computed and to ensure the reservoir did not empty when simulating the 98-year period-of-record. These include the buffer option, previously stated, and an automatic index option. The automatic index option sets the hydrologic index based on the storage in Lake Mendocino. For example, a Critical hydrologic index could be assigned when the storage in Lake Mendocino is below 20,000 acre-feet.

Figure 33 shows the hydrologic index and Lake Mendocino storage from three different simulations. One simulation used only the water balance approach without any tuning knobs. The other two use the buffer option and the buffer plus auto-index options. The

Month	**4**
Beginning	**30000**
Forecast	**1**

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
	Month	Monthly Inflow acre-ft	Loss and Evap acre-ft	Minimum Flow acre-ft	End of Month Storage acre-ft	Water Stored acre-ft	Monthly Usable acre-ft
1	Jan	0	0	0	0	0	0
2	Feb	0	0	0	0	0	0
3	Mar	0	0	0	0	0	0
4	Apr	4,714	1,482	11,008	22,224	0	4,714
5	May	5,561	1,940	11,375	14,470	0	5,561
6	Jun	5,785	4,790	7,438	8,026	0	5,785
7	Jul	5,445	6,350	7,686	0	0	5,445
8	Aug	5,465	5,971	7,686	0	0	5,465
9	Sep	6,537	5,658	7,438	0	0	6,537
10	Oct						
11	Nov						
12	Dec						
						Sum	33,507
						Buffer	30,000
						Total Available Water	33,507
						Total Loss	26,191
						Water Available for Minimum Flow	7,316
						INDEX	**3**

Figure 32. Example Water Balance Spreadsheet Starting in April with Initial Lake Mendocino Storage of 30,000 acre-feet and a "Normal" Forecast.

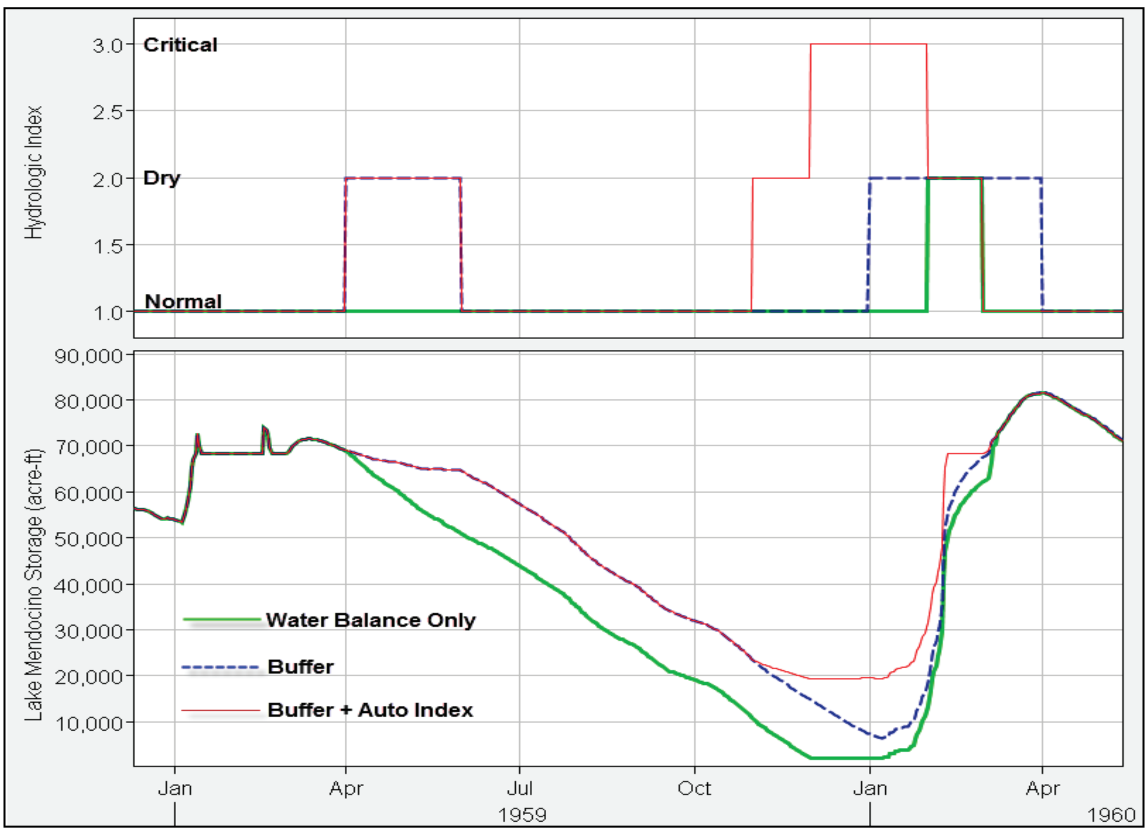

Figure 33. Results from Three of the Water Balance Hydrologic Index Alternatives.

auto-index was set up to switch the index to Dry when the storage in Lake Mendocino dipped below 30,000 acre-feet, and to switch to Critical when it was below 20,000 acre-feet. Notice how the buffer and auto-index options prevent the reservoir from becoming empty during the 1959 water year.

As previously stated, the water balance approach can incorporate a forecast. The forecast dictates inflow into Lake Mendocino and the minimum flow requirement in the Russian River. The forecast option was incorporated into the water balance state variable within the HEC-ResSim model and uses percentiles of historic inflow (the forecast option was designed to be flexible so that other forecast inflows could be used if they became available). For example, the median historic inflow and Normal minimum flow requirement are assumed with a Normal forecast. The twenty-fifth percentile historic inflow and Dry minimum flow requirements are assumed with a Dry forecast, and the tenth percentile historic inflow and Critical minimum flow requirements are assumed with a Critical forecast. For this hydrologic index alternative, the forecast was developed using cumulative inflow into Lake Mendocino (similar to the inflow hydrologic index). This forecast approach generated a forecast of Normal eighty-seven percent of the time, Dry ten percent of the time, and Critical three percent of the time. This approach could easily be modified to use forecasts from meteorologic models.

Table 17 contains the number of Normal, Dry, and Critical periods for the water balance scenarios: Scenario 1 uses the water balance only, Scenario 2 uses the water balance plus buffer option, Scenario 3 uses the water balance plus buffer plus auto-index options, and Scenario 4 uses the water balance plus buffer plus auto-index plus forecast options. These percentages can be modified by adjusting the buffer, auto-index thresholds, or forecast. Model results showed Lake Mendocino going Dry during a number of periods with only the water balance scenario. The buffer, auto-index, and forecast options do prevent the reservoir from drying.

Table 17. Percent Normal, Dry, and Critical for Water Balance Alternatives.

Scenario	Buffer Option	Auto-Index Option	Forecast Option	% Occurrence		
				Normal	Dry	Critical
1				97.2	1.4	1.4
2	X			87.6	8.8	3.6
3	X	X		83.0	13.5	3.5
4	X	X	X	81.1	14.8	4.2

One more option was added to the water supply index, a delay option. The delay option is an attempt to prevent the hydrologic index from bouncing back and forth between different index values (from Normal to Dry and then back to Normal in consecutive months). Figure 34 and Figure 35 show two different types of delay. Figure 34 shows a delay as the hydrologic index is increasing (from Normal to Dry and Critical) and decreasing (back to Normal). In this case, the index is not changed until it is higher or lower than the current index for consecutive periods. Figure 35 shows only a delay as the index is decreasing back to Normal. This approach does not delay changing the index from Normal to Dry or Dry to Critical in order to respond as fast as possible to drier conditions in the watershed.

Table 18 was developed using the water balance spreadsheet to back out the beginning of month storage that results in a Normal, Dry, or Critical index. Beginning storage was computed by assuming either the twenty-fifth percentile inflow or the fiftieth percentile inflow. For example, if assuming the twenty-fifth percentile inflow, Lake Mendocino would need at least 77,000 acre-feet on March 1, or a Critical index would be assigned. Notice how the beginning of each month's thresholds is similar from April through September for both twenty-fifth and fiftieth percentile inflows. This is due to the fact that PVP inflows are consistent during these months (there is very little rainfall-runoff). Also notice the Normal and Dry thresholds are greater than Guide Curve for January, February, and March when

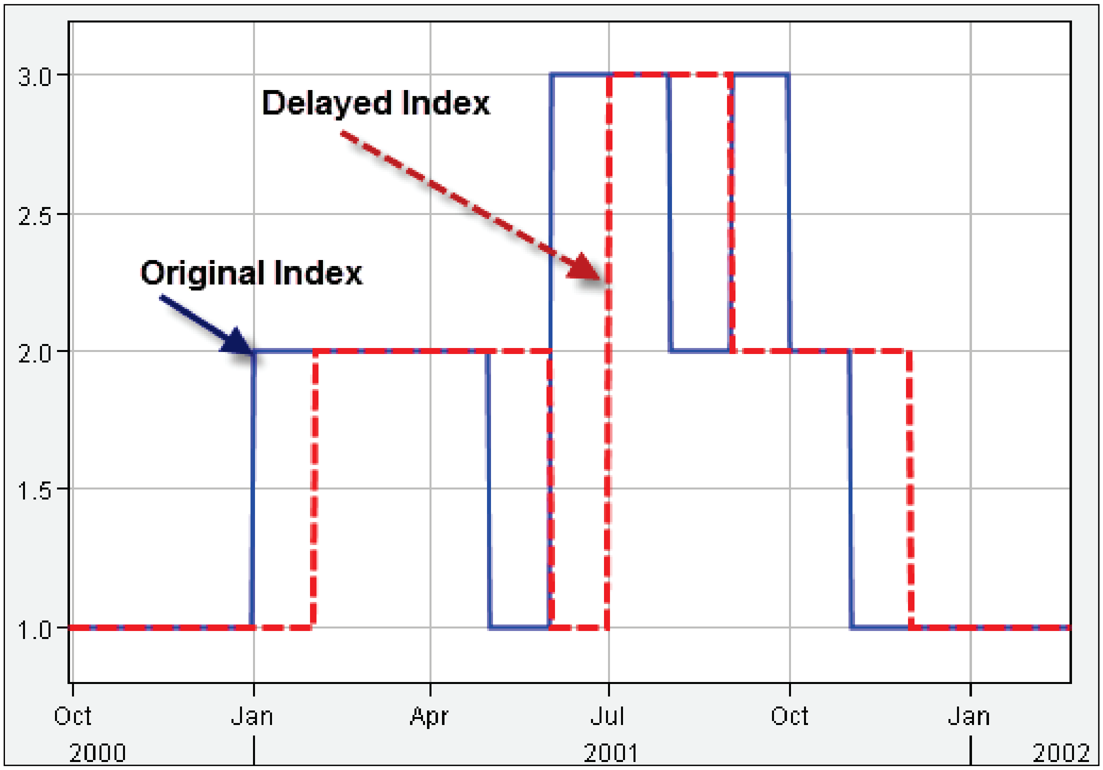

Figure 34. Example of Delaying the Hydrologic Index when it is Increasing and Decreasing.

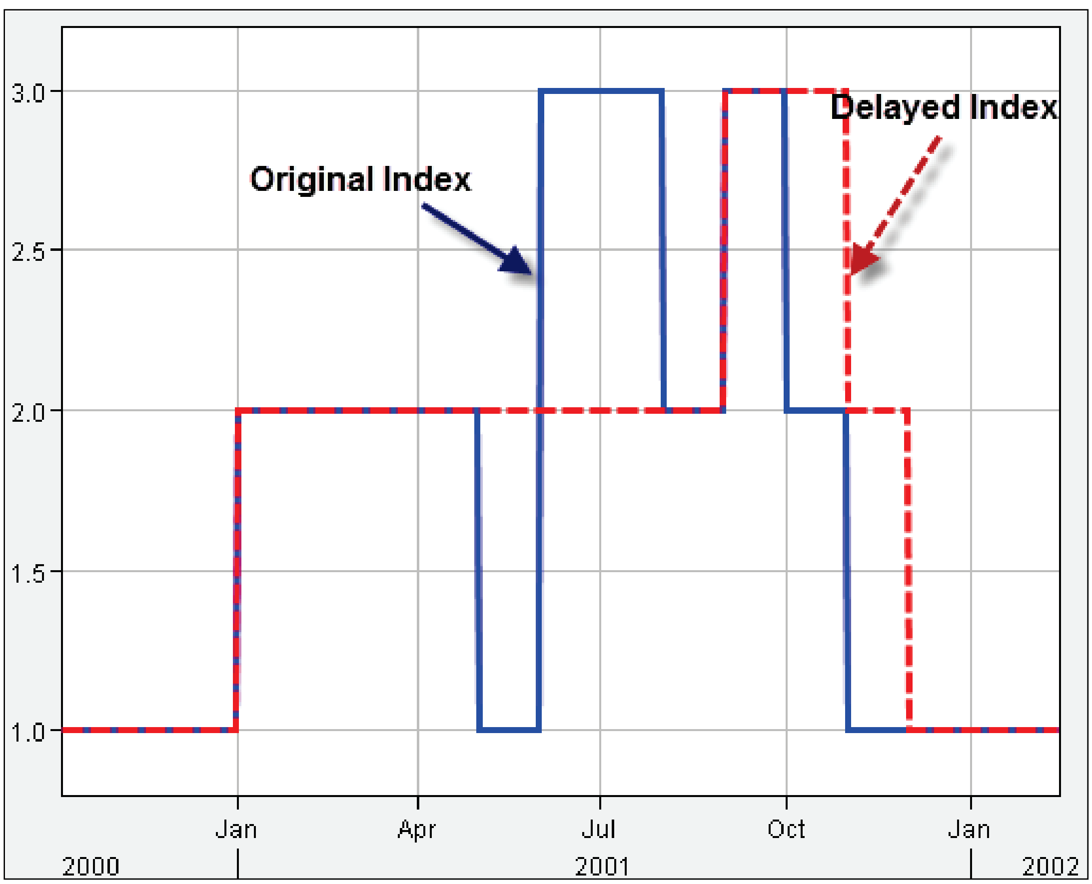

Figure 35. Example of Delaying the Hydrologic Index when it is Decreasing.

Table 18. Operational Table for Computing the Hydrologic Index using the Current Lake Mendocino Storage and Forecast.

	Assume 25th Percentile Inflow				Assume 50th Percentile Inflow		
	Normal	Dry	Critical		Normal	Dry	Critical
	Starting Storage acre-feet	Starting Storage acre-feet	Starting Storage acre-feet	Beginning Month GC acre-feet	Starting Storage acre-feet	Starting Storage acre-feet	Starting Storage acre-feet
Jan	>87,000	<87,000	<17,000	68,409	>22,000	<22,000	----
Feb	>82,000	<82,000	<32,000	68,409	>48,000	<48,000	<14,000
Mar	>77,000	<77,000	<47,000	68,409	>69,000	<69,000	<49,000
Apr	>78,000	<78,000	<53,000	86,400	>76,000	<76,000	<50,000
May	>70,000	<70,000	<51,000	105,930	>67,000	<67,000	<49,000
Jun	>61,000	<61,000	<50,000	110,967	>60,000	<60,000	<48,000
Jul	>56,000	<56,000	<47,000	110,967	>54,000	<54,000	<45,000
Aug	>47,000	<47,000	<41,000	110,967	>45,000	<45,000	<39,000
Sep	>39,000	<39,000	<36,000	110,967	>37,000	<37,000	<34,000

and March when using the twenty-fifth percentile inflows. In most cases these months would be assigned a Dry index if the twenty-fifth percentile inflow was used in the water balance.

4.8 Cumulative Inflow plus Water Balance Hydrologic Index

The cumulative inflow plus water balance hydrologic index uses cumulative inflow into Lake Mendocino to set the hydrologic index from January 1 through May 31 and the water balance approach to set the index from June 1 through December 31. The same cumulative inflow thresholds from the cumulative inflow hydrologic index were used in this analysis (see Table 12). Using the same inflow thresholds, three different scenarios of the water supply approach were simulated with the HEC-ResSim model, a buffer option, a buffer plus auto-index option, and a buffer plus auto-index plus forecast option.

This hydrologic index option approach was added to the HEC-ResSim model as a state variable script (contained in Appendix A.7). During a simulation, the HEC-ResSim script computes the cumulative inflow into Lake Mendocino and the water balance to determine the hydrologic index. The computed hydrologic index was used by HEC-ResSim to set minimum stream flows throughout the Russian River model. Options, tuning knobs, were added to the water balance approach within the HEC-ResSim model to provide flexibility in how many Normal, Dry, and Critical periods were computed and to ensure the reservoir did not empty using the 98-year period-of-record dataset.

Table 19 shows the percent Normal, Dry, and Critical for the three scenarios. The percentages of Normal, Dry, and Critical are changing due to differences in the water supply portion of the index; the inflow thresholds are constant for all three thresholds. As results show, this analysis is not as sensitive to a forecast because reservoir inflows from June 1 – September 30 are fairly uniform. Typically, there is no rainfall-runoff during this period; therefore, a majority of inflow into Lake Mendocino is from the PVP diversion.

Figure 36 shows an example for Scenarios 2 and 3 for 1959 where the forecast improves the analysis. Notice the index values from January 1 through May 31 are the same since both scenarios are using the same cumulative inflows into Lake Mendocino. Differences in

Table 19. Percent Normal, Dry, and Critical from the Cumulative Inflow plus Water Balance Hydrologic Index Alternatives.

	Scenario 1 (buffer)	Scenario 2 (buffer + auto-index)	Scenario 3 (buffer + auto-index + forecast)
Normal Percent	84.9%	79.8%	79.1%
Dry Percent	9.4%	14.8%	15.6%
Critical Percent	5.8%	5.4%	5.3%

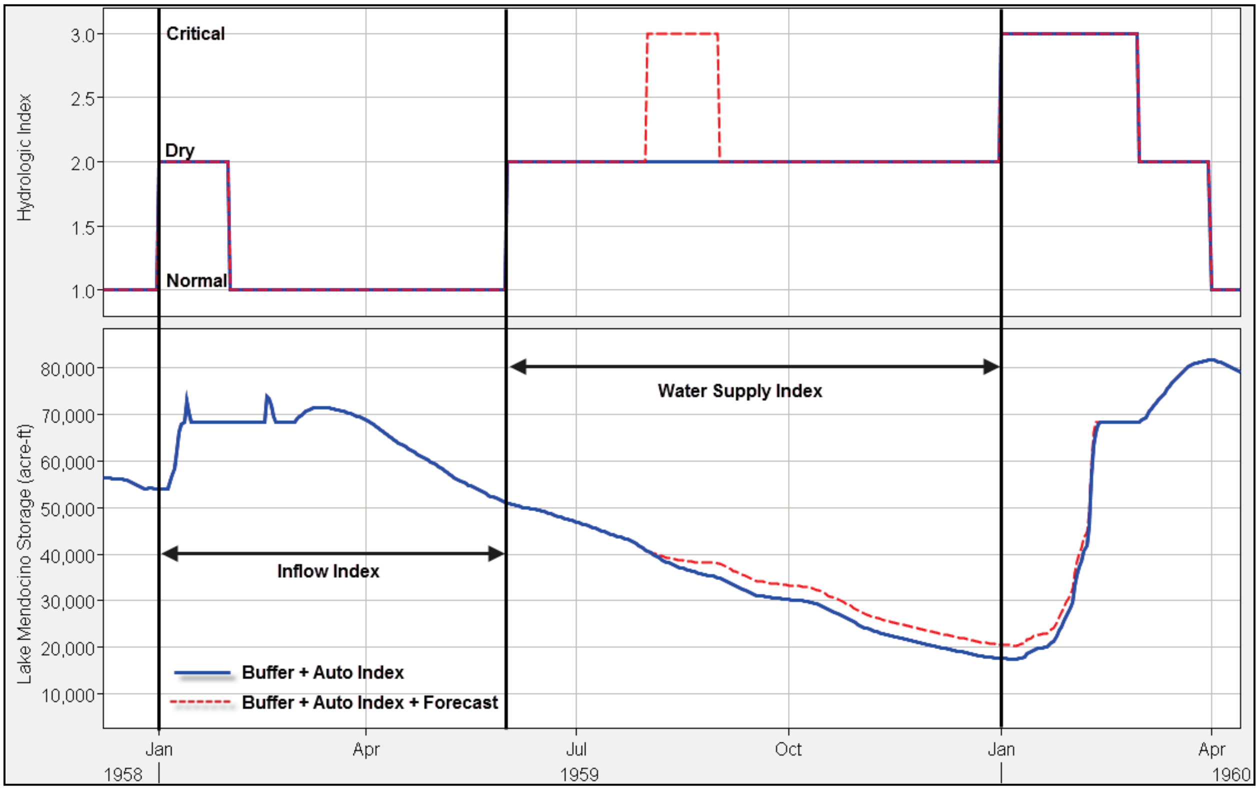

Figure 36. Comparison of Scenarios in the Cumulative Inflow plus Water Balance Hydrologic Index.

the index occur between June 1 and December 31. The buffer plus auto-index plus forecast scenario results in higher storage (less releases), whereas storage for buffer plus auto-index scenario falls below 20,000 acre-feet.

4.9 Water Balance Hydrologic Index with an Ensemble of Forecasted Inflows

The water balance with ensemble option hydrologic index uses the water balance procedure, but instead of using historic/modeled percentiles of inflow into Lake Mendocino, an ensemble approach is utilized. Each time the index is evaluated, actual monthly volumes (inflow into Lake Mendocino) are used by the water balance. These monthly volumes are from the 98-year period-of-record (the unimpaired inflow dataset was converted to volume at a monthly time-step). In effect, the water balance is computed for each historic year using the current reservoir storage. Figure 37 shows monthly inflow volume flowing into Lake Mendocino. This dataset uses the unimpaired flows developed by the USGS and current PVP water dataset.

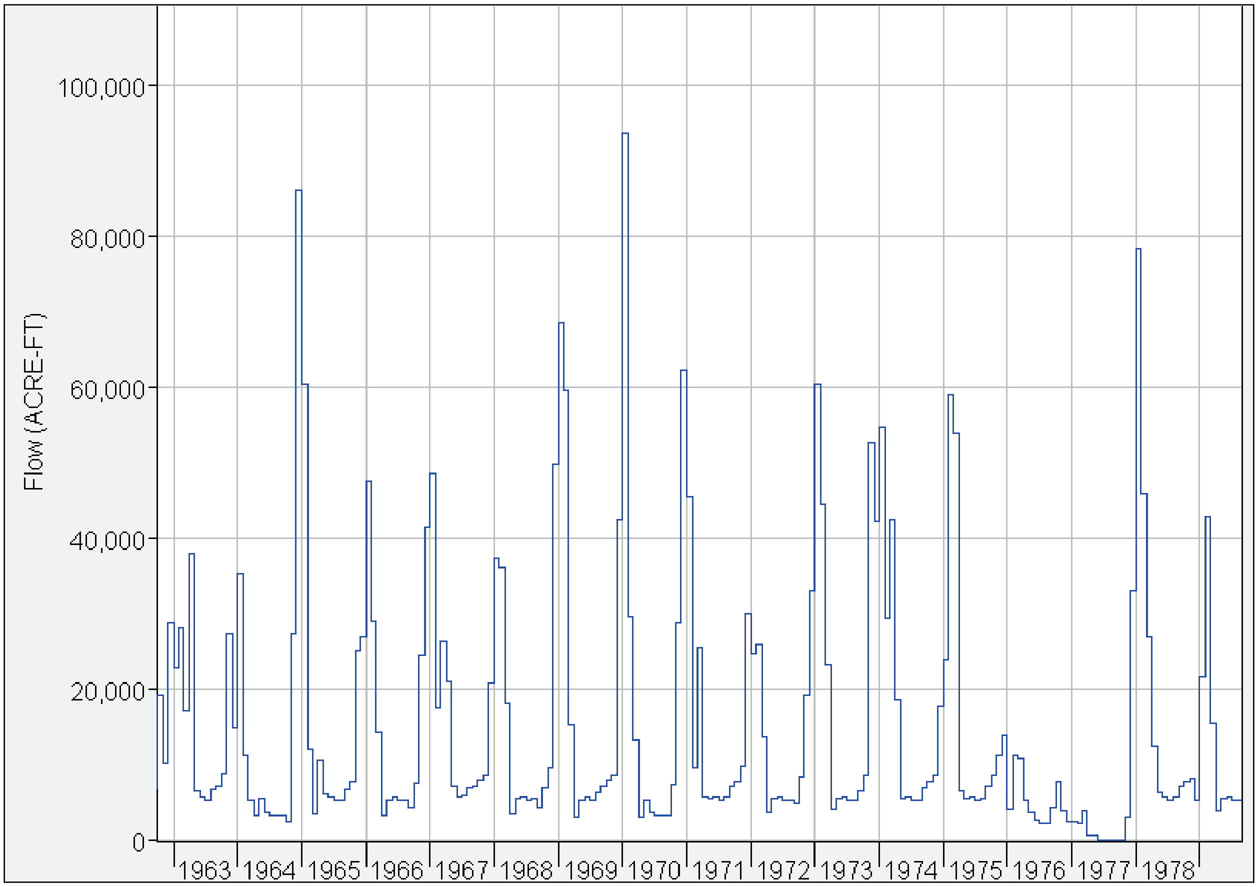

Figure 37. Monthly Inflows (Volume) into Lake Mendocino.

This hydrologic index option was added to the HEC-ResSim model as a state variable script (contained in Appendix A.8). During a simulation, the HEC-ResSim script computes the water available to determine the hydrologic index. The computed hydrologic index was used by HEC-ResSim to set minimum stream flows throughout the Russian River model. The same options, tuning knobs, added to the water balance options were available for this index. The tuning knobs provide flexibility in how many Normal, Dry, and Critical periods were computed during the historic simulation and were used to ensure the reservoir did not empty using the 98-year period-of-record dataset. These include a buffer option, a buffer plus auto-index option, and a buffer plus auto-index plus forecast option. The forecast option is treated differently with this index. Here, the forecast is only used to determine minimum flows for the water balance. As mentioned, forecasted inflows were not used with this hydrologic index option; instead, the water balance is computed 98 times using inflows from each historic year. For example, using the current month and reservoir storage, the water balance is computed using inflows for 1910. Based on this analysis, a hydrologic index is computed. Then, using the same current month and reservoir storage, inflows for 1911 are used to compute another hydrologic index. This is done 98 times, once for each year in the 98-year period-of-record.

The following figures (Figures 38 through 41) show a few examples of how the water balance, with an ensemble of "forecasted" inflows, could be used to compute the hydrologic index. The same simple spreadsheet shown for the water balance index is used to demonstrate the procedure (this logic is included in the HEC-ResSim state variable). Figure 38 shows the input data for the water balance spreadsheet. Notice the input data is similar to Figure 29; however, instead of using the tenth, twenty-fifth, and fiftieth percentile Lake Mendocino inflows, Figure 38 shows inflows from 1944, 1977, and 1998. Only three years from the historic record are shown in the example, and they represent normal (1944), dry (1977), and wet (1998) years. All 98 years of historic Lake Mendocino inflow are used in the HEC-ResSim model. The following examples show how the hydrologic index is computed when starting the analysis on January 1 with 30,000 acre-feet of water in Lake Mendocino (a buffer of 30,000 acre-feet was used). Figure 39 shows that the hydrologic index would be Dry when using inflows from 1944, Figure 40 shows that the hydrologic index would be Critical when using inflows from 1977, and Figure 41 shows that the hydrologic index would be Normal when using inflows from 1998.

Inflow, Losses, and End of Month Guide Curve (Conservation Storage)						
		Forecasted Inflows				EOM
		1944	1977	1998	Loss + Evap	Guide Curve
	Month	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft
1	Jan	24064	2500	80201	618	68409
2	Feb	26667	2370	99090	569	68409
3	Mar	20496	3942	29973	1147	86400
4	Apr	2827	783	14931	1482	105930
5	May	3561	777	6222	1940	110967
6	Jun	2800	124	6071	4790	110967
7	Jul	3391	0	6191	6350	110967
8	Aug	7064	0	7068	5971	110967
9	Sep	7207	81	7208	5658	110967
10	Oct					68409
11	Nov					68409
12	Dec					68409

Volumes Needed for Min Flow for Each Condition						
	Monthly			Cumulative		
	Normal Min	Dry Min	Critical Min	Normal Min	Dry Min	Critical Min
	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft
Jan	9223	4612	1537	79408	40612	13537
Feb	8330	4165	1388	70185	36000	12000
Mar	9224	4612	1538	61855	31835	10612
Apr	11008	4463	1487	52631	27223	9074
May	11375	4611	1537	41623	22760	7587
Jun	7438	4463	1488	30248	18149	6050
Jul	7686	4612	1537	22810	13686	4562
Aug	7686	4611	1537	15124	9074	3025
Sep	7438	4463	1488	7438	4463	1488
Oct						
Nov						
Dec						

Figure 38. Input Data for Water Balance plus Ensemble Spreadsheet.

Month	**1**
Beginning	**35000**
Forcecast	**1944**

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
	Month	Monthly Inflow acre-ft	Loss and Evap acre-ft	Minimum Flow acre-ft	End of Month Storage acre-ft	Water Stored acre-ft	Monthly Usable acre-ft
1	Jan	24,064	618	1,537	56,909	21,909	24,064
2	Feb	26,667	569	1,388	68,409	11,500	13,457
3	Mar	20,496	1,147	1,538	86,220	17,811	20,496
4	Apr	2,827	1,482	1,487	86,078	0	2,827
5	May	3,561	1,940	1,537	86,162	84	3,561
6	Jun	2,800	4,790	1,488	82,684	0	2,800
7	Jul	3,391	6,350	1,537	78,189	0	3,391
8	Aug	7,064	5,971	1,537	77,744	0	7,064
9	Sep	7,207	5,658	1,488	77,805	61	7,207
10	Oct						
11	Nov						
12	Dec						
						Sum	**84,867**
						Buffer	**30,000**
						Total Available Water	**89,867**
						Total Loss	**28,525**
						Water Available for Minimum Flow	**61,342**
						INDEX	**2**

Figure 39. Example Water Balance Spreadsheet Starting in January with an Initial Lake Mendocino Storage of 30,000 acre-feet and Monthly Inflows from 1944.

Month	**1**
Beginning	**35000**
Forcecast	**1977**

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
	Month	Monthly Inflow acre-ft	Loss and Evap acre-ft	Minimum Flow acre-ft	End of Month Storage acre-ft	Water Stored acre-ft	Monthly Usable acre-ft
1	Jan	2500	618	1,537	35,345	345	2,500
2	Feb	2370	569	1,388	35,758	413	2,370
3	Mar	3942	1,147	1,538	37,015	1,257	3,942
4	Apr	783	1,482	1,487	34,829	0	783
5	May	777	1,940	1,537	32,129	0	777
6	Jun	124	4,790	1,488	25,975	0	124
7	Jul	0	6,350	1,537	18,088	0	0
8	Aug	0	5,971	1,537	10,580	0	0
9	Sep	81	5,658	1,488	3,515	0	81
10	Oct						
11	Nov						
12	Dec						
						Sum	**10,577**
						Buffer	**30,000**
						Total Available Water	**15,577**
						Total Loss	**28,525**
						Water Available for Minimum Flow	**0**
						INDEX	**3**

Figure 40. Example Water Balance Spreadsheet Starting in January with an Initial Lake Mendocino Storage of 30,000 acre-feet and Monthly Inflows from 1977.

Month	**1**
Beginning	**35000**
Forecast	**1998**

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
			Loss and		End of Month		
		Monthly Inflow	Evap	Minimum Flow	Storage	Water Stored	Monthly Usable
	Month	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft	acre-ft
1	Jan	80201	618	1,537	68,409	33,409	35,564
2	Feb	99090	569	1,388	68,409	0	1,957
3	Mar	29973	1,147	1,538	86,400	17,991	20,676
4	Apr	14931	1,482	1,487	98,362	11,962	14,931
5	May	6222	1,940	1,537	101,107	2,745	6,222
6	Jun	6071	4,790	1,488	100,900	0	6,071
7	Jul	6191	6,350	1,537	99,204	0	6,191
8	Aug	7068	5,971	1,537	98,764	0	7,068
9	Sep	7208	5,658	1,488	98,826	62	7,208
10	Oct						
11	Nov						
12	Dec						

Sum	**105,888**
Buffer	**30,000**
Total Available Water	**110,888**
Total Loss	**28,525**
Water Available for Minimum Flow	**82,363**
INDEX	**1**

Figure 41. Example Water Balance Spreadsheet Starting in January with an Initial Lake Mendocino Storage of 30,000 acre-feet and Monthly Inflows from 1998.

Results from using an ensemble of Lake Mendocino inflows include 98 separate hydrologic index values. When summing the number of Normal, Dry, and Critical indices from the 98 values, typical results look like forty-five Normal index values, fifty Dry index values, and three Critical index values. The final part of the analysis takes the counts of each hydrologic index and assigns one index for the current month and year being evaluated. The following logic shows how a Critical index would be determined if the count of Critical index values was greater than ten and a Normal index value would be assigned if the number of Normal index values was greater than fifty.

```
Hydrologic Index = Dry
If Critical Count > 10
        Hydrologic Index = Critical
Else If Normal Count > 50
        Hydrologic Index = Normal
```

These count thresholds (counts of Critical, Dry, and Normal) are additional tuning knobs with the ensemble approach. They can be used to control the number of Normal, Dry, and Critical periods. The buffer, auto-index, and forecast options can be used with the ensemble approach (the forecast is only used to determine minimum flows).

Table 20 shows the percentages of Normal, Dry, and Critical indices from three scenarios, buffer, buffer plus auto-index, and buffer plus auto-index plus forecast. The count thresholds were held constant in all three scenarios. These could be adjusted in order to generate more/less Critical and Dry periods. As results show, this analysis is not as sensitive to a forecast because the forecast does not influence inflows into the reservoir.

Table 20. Percent Normal, Dry, and Critical from the Water Balance with the Inflow Ensemble Hydrologic Index Alternative.

	Scenario 1 (buffer)	Scenario 2 (buffer + auto-index)	Scenario 3 (buffer + auto-index + forecast)
Normal Percent	86.1%	81.7%	81.3%
Dry Percent	9.2%	14.2%	14.6%
Critical Percent	4.7%	4.1%	4.1%

Chapter 5
Preferred Alternatives

The hydrologic index alternatives in Chapter 4 were presented to SCWA and after input from SCWA, a list of preferred hydrologic index options were developed. This list included a Lake Mendocino Storage index (evaluated bi-monthly), a future available water index, and a modified storage index. In addition, SCWA wanted to evaluate the preferred hydrologic index alternatives using minimum flows from a three-schedule (used by the current hydrologic index) and five-schedule alternatives. Table 21 contains minimum streamflows in the Upper Russian River for both the three- and five-schedule alternatives (flow values were provided by the SCWA). The methodology for the future available water and modified storage hydrologic index alternatives was developed by SCWA, while CEIWR-HEC assisted in the development of the HEC-ResSim state variable scripts and the development of the thresholds (used to determine number of Normal, Dry, and Critical periods). The thresholds were set by trial and error (model simulations) to recreate a similar number of Normal, Dry, and Critical periods as the current hydrologic index and maintain a minimum storage in Lake Mendocino during the 1977 water year.

Table 21. Three-Schedule and Five-Schedule Minimum Flows for the Upper Russian River.

	Normal Min Flow cfs	Normal-Dry Min Flow cfs	Dry Min Flow cfs	Dry-Critical Min Flow cfs	Critical Min Flow cfs
1-Jan	150	125	75	50	25
1-Apr	185	150	75	50	25
1-Jun	125	100	75	50	25
1-Nov	150	125	75	50	25

Table 22 contains monthly percentages of Normal, Dry, and Critical from the current hydrologic index. These percentages are adjusted for the dry spring and minimum Lake Mendocino storage conditions, which only applies to the upper Russian River. Thresholds were adjusted in these preferred index alternatives to recreate similar percentages of Normal, Dry, and Critical periods per month. The same HEC-ResSim model used to

Table 22. Monthly Percentages of Normal, Dry, and Critical using the Current Hydrologic Index.

	Percentages		
	Normal	Dry	Critical
January	86	8	6
February	79	13	8
March	86	10	4
April	87	10	3
May	85	13	2
June	76	22	2
July	76	22	2
August	76	22	2
September	76	22	2
October	63	35	2
November	64	34	2
December	70	28	2

evaluate the hydrologic index alternatives presented in Chapter 4 was used to evaluate the three preferred hydrologic index alternatives.

5.1 Lake Mendocino Storage Hydrologic Index – Evaluated Bi-Monthly

The Lake Mendocino storage hydrologic index, discussed in Section 4.5, was modified to be evaluated bi-monthly (instead of weekly) and a five-schedule minimum flow alternative was added in addition to the existing three-schedule alternative. The state variable script (contained in Appendix A.9) was designed to compare the current storage in Lake Mendocino to storage thresholds in order to compute the hydrologic index. The hydrologic index is then used to set minimum stream flows.

Figure 42 and Table 23 show the Lake Mendocino storage thresholds for the three- and five-schedule alternatives. For example, if the Lake Mendocino storage is greater than 44,000 acre-feet on October 1, then the hydrologic index is Normal, when using the three- schedule

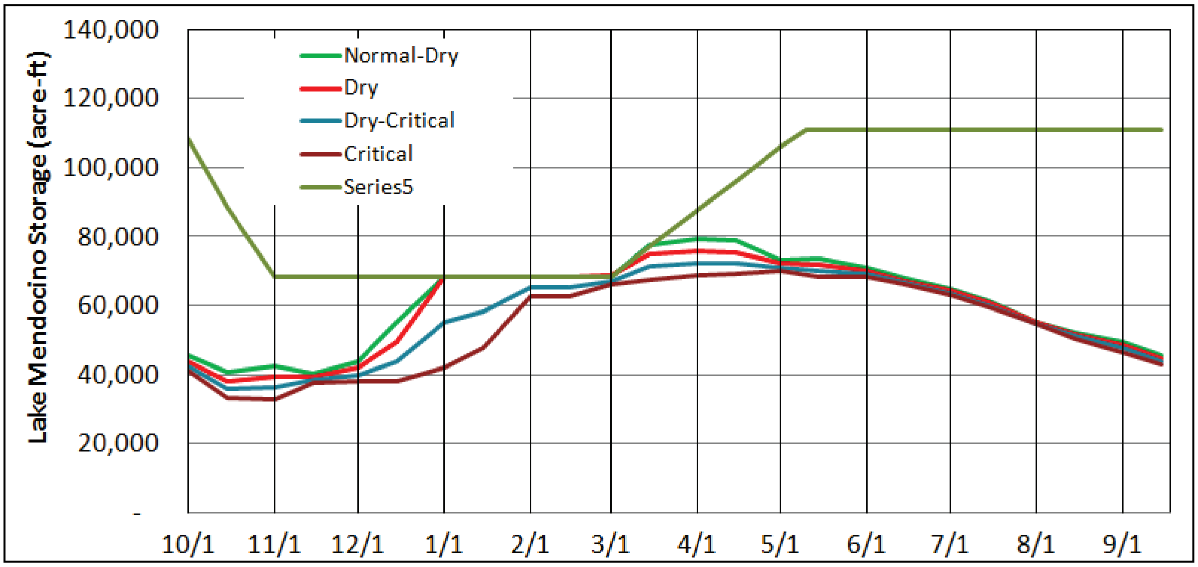

Figure 42. Lake Mendocino Storage Thresholds for the Three- and Five-Schedule Alternatives.

index. When using the five-schedule index, Lake Mendocino storage would need to be greater than 45,350 acre-feet on October 1 in order to have a Normal hydrologic index. Table 24 contains the percent of time over the 98-year simulation that the hydrologic index was Normal, Normal-Dry, Dry, Dry-Critical, and Critical using the storage thresholds in Table 23. Notice the percent of time at a Critical index is higher than the current index, shown in Table 24. This is due to the second constraint for determining storage thresholds, which was to keep Lake Mendocino from going Dry during the 1977 water year. Figure 43 shows a comparison of the modeled hydrologic index and Lake Mendocino storage with the three-schedule versus five-schedule storage thresholds. As shown, there is not much difference in modeled Lake Mendocino storage between the three- and five-schedule alternatives for this one water year (this observation is true for all 98 years).

5.2 Future Available Water Hydrologic Index

An empirical relationship was developed by SCWA relating the hydrologic index to "future" available water for meeting losses and minimum flow in the Upper Russian River. This method is similar to the water balance approach in that it attempts to determine the usable,

Table 23. Lake Mendocino Storage Thresholds for Three- and Five-Schedule Alternatives.

Date	Normal-Dry (acre-feet)	Dry (acre-feet)	Dry-Critical (acre-feet)	Critical (acre-feet)	Guide Curve (acre-feet)
1-Oct	45,350	44,000	42,650	41,300	108,117
15-Oct	40,700	38,300	35,900	33,500	88,572
1-Nov	42,600	39,400	36,200	33,000	68,409
15-Nov	40,350	39,400	38,450	37,500	68,409
1-Dec	44,000	42,000	40,000	38,000	68,409
15-Dec	55,250	49,500	43,750	38,000	68,409
1-Jan	68,300	68,300	55,150	42,000	68,409
15-Jan	68,300	68,300	58,050	47,800	68,409
1-Feb	68,400	68,400	65,450	62,500	68,409
15-Feb	68,400	68,400	65,450	62,500	68,409
1-Mar	68,400	68,600	67,200	66,000	68,409
15-Mar	77,500	75,000	71,200	67,400	77,214
1-Apr	79,300	75,800	72,300	68,800	87,435
15-Apr	78,650	75,500	72,350	69,200	95,997
1-May	73,150	72,100	71,050	70,000	105,930
15-May	73,750	72,000	70,250	68,500	110,967
1-Jun	71,150	70,200	69,250	68,300	110,967
15-Jun	67,800	67,200	66,600	66,000	110,967
1-Jul	64,950	64,300	63,650	63,000	110,967
15-Jul	61,400	60,800	60,200	59,600	110,967
1-Aug	55,350	55,100	54,850	54,600	110,967
15-Aug	52,150	51,600	51,050	50,500	110,967
1-Sep	49,650	48,600	47,550	46,500	110,967
15-Sep	45,400	44,600	43,800	43,000	110,967

Table 24. Percentages of Normal, Dry, and Critical for using the Three- and Five-Schedule Lake Mendocino Storage Hydrologic Index.

	3-Schedule Index			5-Schedule Index				
	Normal	Dry	Critical	Normal	Normal-Dry	Dry	Dry-Critical	Critical
January	68%	23%	10%	68%	0%	8%	14%	9%
February	86%	5%	9%	86%	0%	2%	4%	8%
March	68%	28%	4%	54%	14%	20%	8%	4%
April	71%	21%	8%	56%	17%	13%	8%	6%
May	69%	21%	9%	63%	10%	10%	13%	5%
June	73%	19%	8%	72%	13%	6%	7%	3%
July	70%	23%	7%	61%	23%	6%	8%	2%
August	73%	18%	9%	66%	14%	10%	2%	8%
September	62%	32%	6%	47%	37%	6%	6%	4%
October	72%	21%	7%	61%	15%	10%	12%	2%
November	67%	24%	9%	55%	22%	9%	7%	7%
December	71%	20%	9%	62%	11%	9%	11%	7%

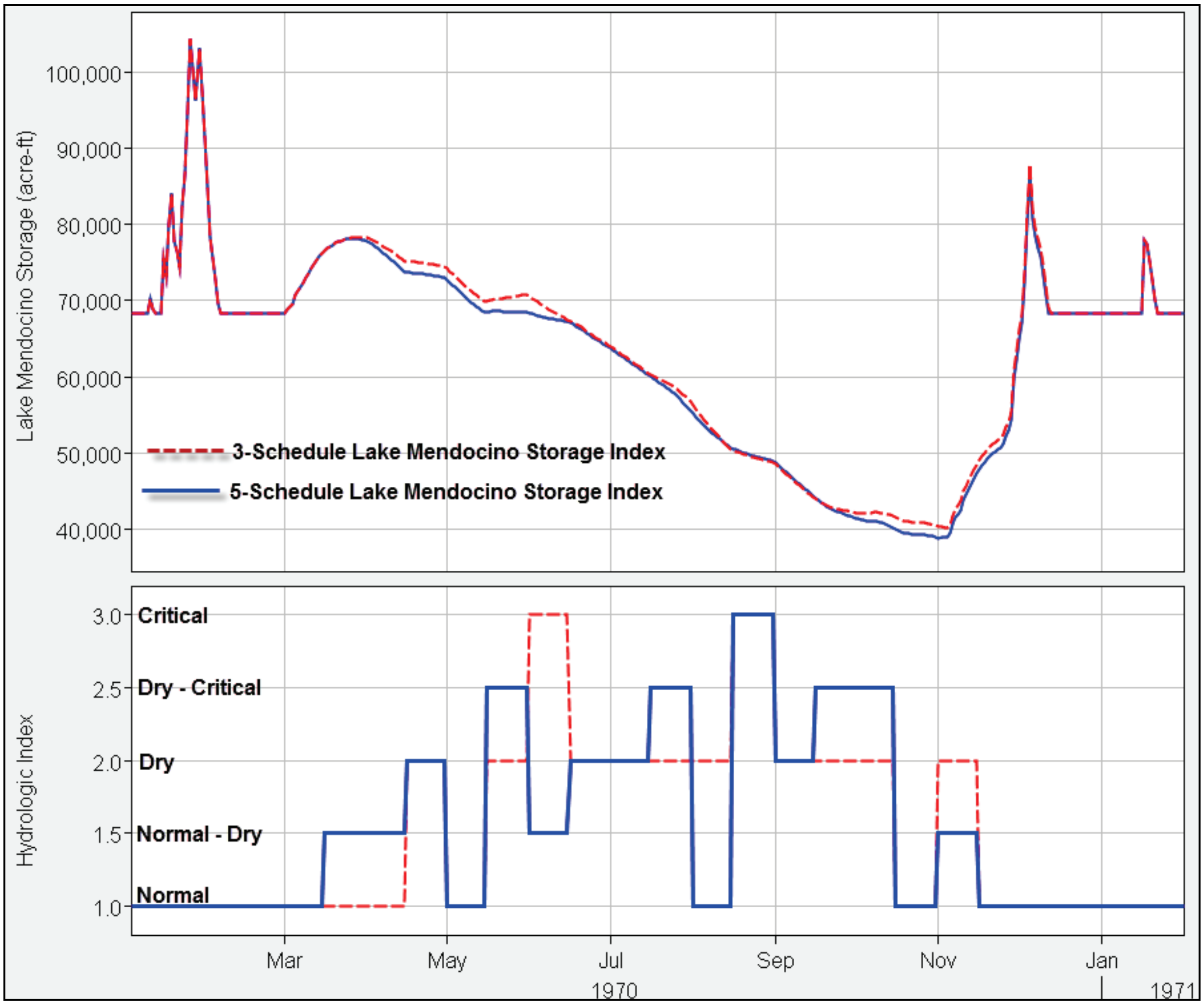

Figure 43. Lake Mendocino Storage and the Hydrologic Index from the Three- and Five-Schedule Storage Alternatives.

or available, water. Not all water flowing into Lake Mendocino can be used for meeting minimum flows and downstream needs. When Lake Mendocino storage is at conservation (guide curve), any additional inflow into the reservoir is passed through the dam to the Russian River (not to exceed downstream maximum flow requirements). This hydrologic index alternative also incorporates a forecasted inflow term, which was designed to use forecasts available from the NWS.

The following equations are used to compute the available water (this is available water from the evaluation point to the end of the water year).

PCI_{p1} is the predicted cumulative inflow into Lake Mendocino from January 1 through May 15 (wet season).
PCD_{p1} is the predicted PVP diversions into the East Branch Russian River (Period 1).
PCD_{p1} values were estimated using modeled PVP diversions designed to approximate current operations.
The A variable is a multiplier used to weight forecasted inflows for Period 1.

As shown, the A variable is a function of current storage in Lake Mendocino. If the Lake Mendocino Current Storage (LMCS) is lower than the conservation or guide curve storage, (Lake Mendocino Maximum Storage, LMMS), then Lake Mendocino is able to store more of the predicted inflows and use them to meet downstream needs (the A variable is larger). If the current Lake Mendocino storage is at the conservation storage (the reservoir is full) then the A variable will be rather small because not all inflow is usable; reservoir inflow will be passed downstream. The C and B terms were developed by SCWA through a regression analysis using modeled output, observed inflows and PVP water, and the available water equation. Table 25 contains the input data for the future available water index. The available water computed with this empirical equation is compared to thresholds in order to define the hydrologic index. These available water thresholds were determined by

Table 25. Input Data used for the Future Available Water Hydrologic Index Option.

	Beginning of Period					PCD Period 1 (acre-ft)			PCD Period 2 (acre-ft)		
Period	Month	Day	B	C	LMMS	Normal	Dry	Critical	Normal	Dry	Critical
1	January	1	14,560	1.08	68,409	24,180	18,360	4,660	43,496	30,169	6,212
	January	16	11,200	1.16	68,409	19,836	16,036	4,232	43,496	30,169	6,212
	February	1	12,060	1.19	68,409	15,203	13,048	2,884	43,496	30,169	6,212
	February	15	9,985	1.29	68,409	11,149	9,744	2,221	43,496	30,169	6,212
	March	1	8,660	1.43	68,409	7,094	6,045	963	43,496	30,169	6,212
	March	16	4,359	1.52	77,214	4,950	4,315	815	43,496	30,169	6,212
	April	1	1,811	1.59	86,828	3,203	2,599	446	43,496	30,169	6,212
	April	16	180	1.83	95,997	2,162	1,575	298	43,496	30,169	6,212
	May	1	180	1.91	105,305	1,121	802	149	43,496	30,169	6,212
2	May	16	180	0	110,967	0	0	0	43,496	30,169	6,212
	June	1	54	0	110,967	0	0	0	41,116	29,131	6,054
	June	16	13	0	110,967	0	0	0	38,933	28,388	5,905
	July	1	0	0	110,967	0	0	0	36,750	27,644	5,756
	July	16	0	0	110,967	0	0	0	34,567	26,900	5,607
	August	1	0	0	110,967	0	0	0	32,160	25,849	5,449
	August	16	0	0	110,967	0	0	0	29,787	24,689	5,300
	September	1	0	0	110,967	0	0	0	27,147	23,237	5,141
	September	16	0	0	110,967	0	0	0	24,582	21,877	4,993
	October	1	0	0	108,117	0	0	0	22,289	19,597	4,844
	October	16	0	0	87,200	0	0	0	19,697	16,949	4,695
	November	1	0	0	68,409	0	0	0	15,517	12,824	4,536
	November	16	0	0	68,409	0	0	0	11,451	9,271	4,388
	December	1	0	0	68,409	0	0	0	7,623	6,154	4,239
	December	16	0	0	68,409	0	0	0	3,908	3,251	3,251

simulating the HEC-ResSim model with the goal of recreating a similar number of Normal, Dry, and Critical periods as the current hydrologic index and maintaining a minimum storage in Lake Mendocino during the 1977 water year.

$$\text{Available Water} = A \times (PCI_{P1} + PCD_{P1}) + B + PCD_{P2} + LMCS$$

$$A = C - \frac{LMCS}{LMMS}$$

If A > 1 then: A=1

Definition of Terms:

Available Water = estimated future available water (acre-feet).
A = Period 1 inflow multiplier.
B = defined in input table.
C = defined in input table.
LMCS = Lake Mendocino current storage (acre-feet).
LMMS = Lake Mendocino maximum storage as defined by guide curve (acre-feet).
BOP = beginning of period.
PCI_{P1} = predicted Lake Mendocino cumulative unimpaired inflow for Period 1 (acre-feet). If BOP is after May 15 then PCI_{P1}=0.
PCD_{P1} = predicted PVP cumulative diversion for Period 1. If BOP is after May 15, then PCD_{P1}=0.

PCD_{P2} = predicted PVP cumulative diversion for Period 2. If BOP is after May 15, then Period 2 is defined by BOP to December 31.

The input data and available water thresholds, developed by SCWA, were input into the state variable script in the HEC-ResSim model. This alternative hydrologic index was designed to be evaluated bi-monthly (instead of weekly), and three-schedule and five-schedule minimum flow alternatives were developed. The state variable script (contained in Appendix A.10) will compare the available water computed using the input data in Table 25, current Lake Mendocino Storage, and predicted inflows (from the NWS and assumed PVP water), to the thresholds (contained in Table 26) in order to compute the hydrologic index. The hydrologic index is then used to set minimum streamflows.

Table 26. Available Water Thresholds for Three- and Five-Schedule Alternatives.

Date	Normal (acre-feet)	Normal-Dry (acre-feet)	Dry (acre-feet)	Dry-Critical (acre-feet)	Critical (acre-feet)
1-Oct	141,855	140,437	135,990	132,721	128,707
15-Oct	142,047	138,662	132,420	128,786	124,540
1-Nov	138,042	136,595	130,532	125,525	119,145
15-Nov	131,152	130,585	128,429	120,621	116,008
1-Dec	130,775	129,877	124,181	118,226	111,560
15-Dec	124,203	122,657	119,663	113,736	107,850
1-Jan	118,134	117,972	115,697	113,951	111,460
15-Jan	117,283	117,142	116,221	114,163	111,127
1-Feb	117,781	117,498	115,407	114,391	111,292
15-Feb	116,987	116,866	115,832	112,996	108,715
1-Mar	117,466	116,708	113,162	110,938	107,657
15-Mar	115,316	114,825	112,291	109,351	106,428
1-Apr	115,607	114,805	112,632	109,502	105,861
15-Apr	113,873	113,518	111,383	107,085	101,687
1-May	111,145	110,258	107,395	102,101	96,155
15-May	105,717	104,623	101,352	96,507	89,822
1-Jun	99,790	98,997	95,270	90,065	83,089
15-Jun	94,142	93,373	89,830	84,249	77,331
1-Jul	87,783	86,965	83,855	78,584	71,109
15-Jul	81,129	80,557	77,289	72,398	64,898
1-Aug	73,275	73,088	69,473	64,615	57,269
15-Aug	66,355	65,800	62,446	57,915	50,561
1-Sep	59,610	59,513	55,642	51,826	44,213
15-Sep	52,482	52,370	49,610	45,092	37,595

This hydrologic index alternative was designed around the use of forecasted flows from NWS. This study used the new ensemble forecasts for the Russian River from NWS. The ensemble forecasts generate thirty-six different possible flow forecasts using the current state of the watershed, represented by a hydrologic model, and applying meteorologic forecasts in the near term, out to two weeks, and then historic precipitation and temperature from the historic years (1961 – 1997). Ensemble forecasts (hindcasts) from 1981 through 2008 were generated from this methodology. This dataset was incorporated into the HEC-ResSim model's state variable script by taking percentiles of the ensembles. For example, a dataset of the fiftieth percentile forecasted flow was created from the ensemble forecast. Other percentiles could be incorporated into this index.

After evaluation of the ensemble forecasts and discussion with the NWS, it was found that the forecasts were not at a good enough quality for use in the hydrologic index, and discussion with NWS indicated the ensemble forecast were not likely to improve in the near future for the Russian River watershed. Figure 44 shows a comparison of the unimpaired inflows from the USGS model and the tenth, thirtieth, and fiftieth percentile inflows from the ensemble forecasts. Notice the forecasts do not change much from year to year. Runoff in the Russian River watershed is driven by rain, not snowmelt. Therefore, an ensemble that can only incorporate historic precipitation will show little variation in the ensemble forecast from one year to the next.

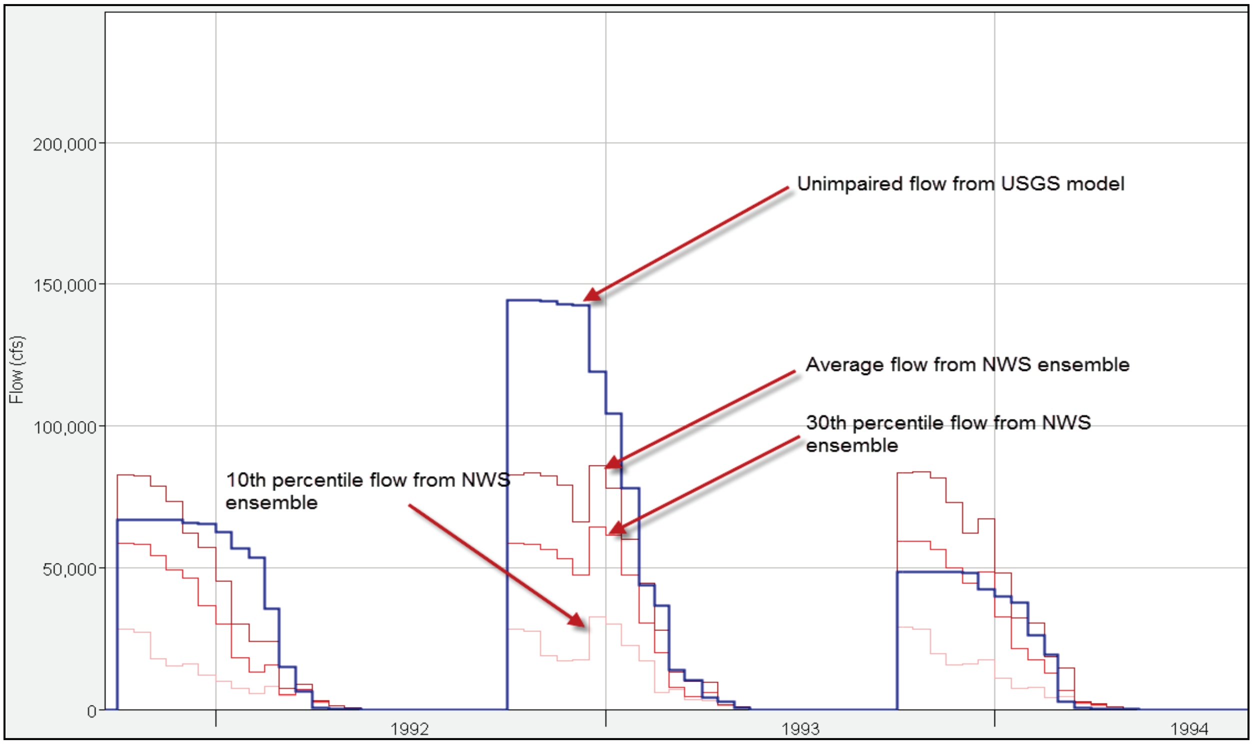

Figure 44. Comparison of National Weather Service Ensemble Forecasts to Unimpaired Flows from the USGS Model.

Table 26 shows the available water thresholds for the three-schedule and five-schedule alternatives. Table 27 contains the percent of time over the 98-year simulation that the hydrologic index was Normal, Normal-Dry, Dry, Dry-Critical, and Critical, using the available water thresholds in Table 26. Notice the percent of time at a Critical index is higher than the current index, shown in Table 22. This is due to the second constraint for determining thresholds, which was to keep Lake Mendocino from going dry during the 1977 water year.

5.3 Modified Storage Hydrologic Index

The modified storage hydrologic index was developed by SCWA. This alternative hydrologic index was designed to be evaluated bi-monthly and three- and five-schedule minimum flow alternatives were developed. Instead of evaluating storage in Lake Mendocino only, the modified storage index looks at Lake Mendocino storage plus predicted PVP water plus the bi-monthly flow at the West Fork of the Russian River USGS stream gage from the previous evaluation period (see the equation below). The state variable script (contained in Appendix A.11) was designed to compare the combined storage from these three sources to storage

Table 27. Percentages of Normal, Dry, and Critical for using the Three- and Five-Schedule Future Available Water Index.

	3-Schedule Index			5-Schedule Index				
	Normal	**Dry**	**Critical**	**Normal**	**Normal-Dry**	**Dry**	**Dry-Critical**	**Critical**
January	83%	9%	8%	58%	27%	5%	3%	7%
February	70%	21%	10%	47%	33%	8%	4%	8%
March	69%	25%	6%	49%	35%	7%	4%	5%
April	70%	21%	9%	57%	21%	11%	6%	5%
May	68%	22%	10%	49%	28%	10%	7%	7%
June	71%	20%	9%	61%	19%	8%	4%	7%
July	69%	21%	10%	59%	18%	12%	5%	6%
August	69%	23%	8%	67%	14%	11%	3%	5%
September	76%	19%	5%	71%	6%	17%	3%	4%
October	68%	23%	10%	64%	13%	9%	12%	3%
November	71%	18%	11%	61%	17%	9%	5%	9%
December	68%	19%	13%	60%	12%	10%	12%	6%

thresholds in order to compute the hydrologic index. The hydrologic index is then used to set minimum stream flows.

Modified Storage = LMCS + WFI + PMI

Definition of Terms:

LMCS = current Lake Mendocino storage (acre-feet).
PMI = predicted PVP minimum cumulative diversions from April 1 through September 30, as determined by the PVP hydrologic condition (acre-feet). These values are defined in Table 28.
WFI = bi-monthly flow from the West Fork of the Russian River USGS gage, 114610000 from the previous evaluation period (acre-feet).

For instance if the evaluation date was January 1, the index would be calculated by adding LMCS plus WFI (the total flow volume from December 16 to the 31^{st}) and the PMI term would be added as provided in Table 28.

Table 29 shows the Lake Mendocino storage thresholds for the three-schedule and five-schedule alternatives. Table 30 contains the percent of time over the 98-year simulation the hydrologic index was Normal, Normal-Dry, Dry, Dry-Critical, and Critical, using the modified storage thresholds in Table 29.

Table 28. Input Data for the Modified Storage Hydrologic Index.

Period	Beginning of Period Month	Day	PMI (acre-ft) PVP Normal	PVP Dry	PVP Critical
1	October	1	0	0	0
2	October	16	0	0	0
3	November	1	0	0	0
4	November	15	0	0	0
5	December	1	0	0	0
6	December	16	0	0	0
7	January	1	0	0	0
8	January	16	0	0	0
9	February	1	0	0	0
10	February	16	0	0	0
11	March	1	0	0	0
12	March	16	0	0	0
13	April	1	22,542	9,650	1,815
14	April	16	21,501	8,628	1,666
15	May	1	20,460	7,884	1,517
16	May	16	19,339	7,141	1,369
17	June	1	16,959	6,347	1,210
18	June	16	14,727	5,603	1,061
19	July	1	12,496	4,860	912
20	July	16	10,265	4,116	764
21	August	1	7,884	3,322	605
22	August	16	5,653	2,579	456
23	September	1	3,273	1,785	298
24	September	16	1,041	1,041	149

Table 29. Storage Thresholds for Three- and Five-Schedule Modified Storage Index.

Date	Normal-Dry (acre-feet)	Dry (acre-feet)	Dry-Critical (acre-feet)	Critical (acre-feet)
1-Oct	45,050	42,200	39,350	36,500
15-Oct	44,000	40,000	36,000	32,000
1-Nov	44,000	40,000	36,000	32,000
15-Nov	41,250	39,000	36,750	34,500
1-Dec	41,500	39,000	36,500	34,000
15-Dec	46,000	43,000	40,000	37,000
1-Jan	51,500	48,000	44,500	41,000
15-Jan	75,250	66,500	57,750	49,000
1-Feb	82,050	72,200	62,350	52,500
15-Feb	83,050	77,500	71,950	66,400
1-Mar	81,700	77,800	73,900	70,000
15-Mar	81,100	77,400	73,700	70,000
1-Apr	88,000	82,000	76,000	70,000
15-Apr	109,600	101,400	93,200	85,000
1-May	102,600	97,800	93,000	87,500
15-May	102,600	97,800	93,000	87,500
1-Jun	99,100	94,400	89,700	85,000
15-Jun	92,400	89,600	86,000	84,000
1-Jul	86,050	84,500	83,400	81,400
15-Jul	84,400	83,000	81,600	80,200
1-Aug	81,200	79,775	78,350	76,925
15-Aug	78,000	76,550	75,100	73,650
1-Sep	74,800	73,325	71,850	70,375
15-Sep	71,600	70,100	68,600	67,100

Table 30. Percentages of Normal, Dry, and Critical for using the Three-Schedule and Five-Schedule Modified Storage Hydrologic Index.

	3-Schedule Index			5-Schedule Index				
	Normal	**Dry**	**Critical**	**Normal**	**Normal-Dry**	**Dry**	**Dry-Critical**	**Critical**
January	69.1%	19.0%	11.9%	53.7%	15.4%	9.9%	10.2%	10.8%
February	68.8%	23.0%	8.2%	59.5%	9.3%	12.5%	10.0%	8.7%
March	69.6%	22.6%	7.8%	55.3%	14.4%	9.9%	13.2%	7.2%
April	69.5%	22.9%	7.6%	51.8%	19.0%	15.1%	7.0%	7.1%
May	71.1%	20.3%	8.6%	52.5%	26.4%	9.2%	4.4%	7.5%
June	71.7%	21.2%	7.1%	69.7%	17.2%	4.0%	4.0%	5.1%
July	71.7%	21.2%	7.1%	69.7%	17.2%	4.0%	4.0%	5.1%
August	71.7%	21.2%	7.1%	69.7%	17.2%	4.0%	4.0%	5.1%
September	71.5%	21.1%	7.4%	69.0%	17.0%	4.5%	4.4%	5.1%
October	68.2%	21.4%	10.4%	46.6%	17.5%	19.9%	9.9%	6.1%
November	67.2%	23.2%	9.6%	58.0%	15.2%	12.1%	7.8%	6.9%
December	73.5%	16.1%	10.4%	68.3%	7.1%	7.0%	8.9%	8.7%

Chapter 6
Conclusion

The purpose of this study was to evaluate hydrologic index alternatives for determining hydrologic conditions (index) and setting the minimum instream flows for the Russian River system. A number of hydrologic index alternatives were evaluated using a HEC-ResSim model of the Russian River. Boundary condition data for the HEC-ResSim model were provided by SCWA and include unimpaired hydrographs throughout the Russian River Watershed, diversions from the Eel River into the Russian River, and distributed losses (evapotranspiration by riparian vegetation, aquifer recharge, agricultural diversions, and non-SCWA municipal and industrial (M&I) diversions) throughout the Russian River watershed. The hydrologic index alternatives were evaluated by simulating a 98-year period-of-record (1910 – 2008).

The hydrologic index alternatives used hydrologic data that were specific for the Russian River watershed, such as cumulative inflow into Lake Mendocino, storage in Lake Mendocino, and a water balance of the Upper Russian River. A few of the hydrologic index alternatives were developed with the ability to incorporate forecasted inflows into Lake Mendocino from NWS. The model alternatives were designed to compute the hydrologic index by evaluating a number of conditions in the model, such as cumulative inflow and storage in Lake Mendocino.

Just like the current hydrologic index, these new alternatives included thresholds that are used to determine hydrologic index (i.e. Normal, Dry and Critical index). These thresholds are part of the model alternative and can be modified to change the percent of time that Normal, Dry, and Critical conditions occur during the 98-year period-of-record HEC-ResSim simulation. This study showed that the thresholds could be modified by trial and error (model simulations) to recreate a similar number of Normal, Dry, and Critical periods as the current hydrologic index and to minimize the risk of Lake Mendocino from going dry during the 98-year period-of-record simulation.

The hydrologic index model alternatives developed as part of this study were provided to SCWA for their further development. SCWA developed a matrix of summary metrics, such as the number of years the ending storage in Lake Mendocino was less than 36,000 acre-feet, average minimum storage, and a metric that describes how often the hydrologic condition changes from one hydrologic condition to another, to help narrow and refine a preferred hydrologic index alternative. Based on analysis by SCWA, their preferred hydrologic index is a five-schedule minimum flow alternative that evaluates cumulative inflow into and storage in Lake Mendocino. SCWA was able to develop a model alternative for the HEC-ResSim model that computes the hydrologic index and sets minimum flows. Using the HEC-ResSim model, SCWA modified cumulative inflow and storage thresholds in order for the percentages of hydrologic conditions to closely approximate percentages observed from the current hydrologic index. The next step is for SCWA to present their preferred hydrologic index alternative to the California State Water Resources Control Board for approval and implementation.

Chapter 7
References

USACE, 1984. "*Warm Springs Dam and Lake Sonoma Dry Creek*", California Water Control Manual. U.S. Army Corps of Engineers, Sacramento District.

USACE, 1986. "*Coyote Valley Dam and Lake Mendocino Russian River*", California Water Control Manual. U.S. Army Corps of Engineers, Sacramento District.

USACE, 2003. "*Coyote Valley Dam and Lake Mendocino Russian River*", California Water Control Manual. U.S. Army Corps of Engineers, Sacramento District.

Appendix A State Variable Scripts

A.1 Lake Mendocino Cumulative Inflow Hydrologic Index

The state variable code (script) for the Lake Mendocino cumulative inflow hydrologic index is shown below. This script is evaluated by HEC-ResSim each time-step during the model simulation. Output from the script is the hydrologic index (referred to as YearType). HEC-ResSim uses the hydrologic index when setting minimum flows from Lake Mendocino and Lake Sonoma. The "initialization" and "clean up" section of the state variable script have been omitted and only the computation section is included below.

```
#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####

from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
year = curTime.year()
mon = curTime.month()
day = curTime.day()
curStepMon = currentRuntimestep.month()

#Set simulation start time, year, month, and day variables
startTime = currentRuntimestep.getRunTimeWindow().getStartTime()
endTime = currentRuntimestep.getRunTimeWindow().getEndTime()
styear = startTime.year()
stmon = startTime.month()
stday = startTime.day()

#Set date for the beginning of the Water Year
if mon >= 10:
	datestr = "1OCT" + str(year)
else:
	datestr = "1OCT" + str(year-1)

bgnWY = HecTime()
bgnWY.setDate(datestr)
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime)
CummPeriod = curTime.julian() - bgnWY.julian() + 1

#Create Lake Mendoncino Inflow Time Series object
LMinflowTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Flow-IN")

#------------------------------Calculate Cumulative Flow----------------------------------
#Get previous value from slave variable
LMCummInflowSV = network.getStateVariable("slave_LMCummInflow")
LMCummInflowTS = LMCummInflowSV.getTimeSeries()
```

```
#Get previous cumulative Lake Mendocino Inflow
if mon == 10 and day ==1:
        prevLMCummInflow = 0
else:
        prevLMCummInflow = LMCummInflowTS.getPreviousValue(currentRuntimestep)

#Convert to acre-feet/day
curLMInflow = LMinflowTS.getValue(step+1)*1.9835

#Calculate Cumulative Inflow
curLMCummInflow = prevLMCummInflow + curLMInflow
LMCummInflowSV.setValue(currentRuntimestep, curLMCummInflow)
#-------------------------------------------------------------------------------------

#Initialize YearType (Hydrologic Index)
YearType = currentVariable.getPreviousValue(currentRuntimestep)

#Evaluate Lake Mendocino Cumulative Inflow (compare to thresholds) to set the
#Hydrologic Index (first of the month)
#January
if mon == 1 and day ==1:
        if curLMCummInflow < 19250:
                YearType = 3
        elif curLMCummInflow < 22500 and curLMCummInflow >= 19250:
                YearType = 2
        else:
                YearType = 1

#February
if mon == 2 and day ==1:
        if curLMCummInflow < 37000:
                YearType = 3
        elif curLMCummInflow < 54000 and curLMCummInflow >= 37000:
                YearType = 2
        else:
                YearType = 1
#March
if mon == 3 and day ==1:
        if curLMCummInflow < 49000:
                YearType = 3
        elif curLMCummInflow < 72000 and curLMCummInflow >= 49000:
                YearType = 2
        else:
                YearType = 1
#April
if mon == 4 and day ==1:
        if curLMCummInflow < 54000:
                YearType = 3
        elif curLMCummInflow < 93500 and curLMCummInflow >= 54000:
                YearType = 2
        else:
                YearType = 1
```

```
#May
if mon == 5 and day ==1:
	if curLMCummInflow < 54000:
		YearType = 3
	elif curLMCummInflow < 99000 and curLMCummInflow >= 54000:
		YearType = 2
	else:
		YearType = 1
#June
if mon == 6 and day ==1:
	if curLMCummInflow < 54000:
		YearType = 3
	elif curLMCummInflow < 102000 and curLMCummInflow >= 54000:
		YearType = 2
	else:
		YearType = 1

#Save Hydrologic Index
currentVariable.setValue(currentRuntimestep, YearType)
```

A.2 Lake Mendocino Storage Hydrologic Index

The state variable code (script) for the Lake Mendocino storage hydrologic index is shown below. This script is evaluated by HEC-ResSim each time-step during the model simulation. Output from the script is the hydrologic index (referred to as YearType). HEC-ResSim uses the hydrologic index when setting minimum flows from Lake Mendocino and Lake Sonoma. The "initialization" and "clean up" section of the state variable script have been omitted and only the computation section is included below.

```
#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####

from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
year = curTime.year()
mon = curTime.month()
day = curTime.day()

#Set simulation start time, year, month, and day variables
startTime = currentRuntimestep.getRunTimeWindow().getStartTime()
endTime = currentRuntimestep.getRunTimeWindow().getEndTime()
styear = startTime.year()
stmon = startTime.month()
stday = startTime.day()

#Set date for the beginning of the Water Year
if mon >= 10:
	datestr = "1OCT" + str(year)
```

```
else:
        datestr = "1OCT" + str(year-1)

bgnWY = HecTime()
bgnWY.setDate(datestr)
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime)
CummPeriod = curTime.julian() - bgnWY.julian() + 1

#Initialize YearType (Hydrologic Index)
YearType = currentVariable.getPreviousValue(currentRuntimestep)

#Evaluate Lake Mendocino Storage (compare to thresholds) to set the
#Hydrologic Index (first of the month)
#Storage threshold is calculated using current Lake Mendocino storage divided by
#guide curve storage
#January
if mon == 1 and day ==1:
        LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
        curLMStorage = LMstorageTS.getValue(step+1)
        LMstorageConTS = network.getTimeSeries("Reservoir","Lake Mendocino",
                "Conservation", "Stor-ZONE")
        curLMStorageCon = LMstorageConTS.getValue(step+1)
        storRatio = curLMStorage/curLMStorageCon
        if storRatio > 0.59:
                YearType = 1
        elif storRatio > 0.50:
                YearType = 2
        else:
                YearType = 3
#February
if mon == 2 and day ==1:
        LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
        curLMStorage = LMstorageTS.getValue(step+1)
        LMstorageConTS = network.getTimeSeries("Reservoir","Lake Mendocino",
                "Conservation", "Stor-ZONE")
        curLMStorageCon = LMstorageConTS.getValue(step+1)
        storRatio = curLMStorage/curLMStorageCon
        if storRatio > .99:
                YearType = 1
        elif storRatio > 0.77:
                YearType = 2
        else:
                YearType = 3
#March
if mon == 3 and day ==1:
        LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
        curLMStorage = LMstorageTS.getValue(step+1)
        LMstorageConTS = network.getTimeSeries("Reservoir","Lake Mendocino",
                "Conservation", "Stor-ZONE")
        curLMStorageCon = LMstorageConTS.getValue(step+1)
        storRatio = curLMStorage/curLMStorageCon
        if storRatio > .99:
                YearType = 1
        elif storRatio > 0.85:
```

```
		YearType = 2
	else:
		YearType = 3
#April
if mon == 4 and day ==1:
	LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
	curLMStorage = LMstorageTS.getValue(step+1)
	LMstorageConTS = network.getTimeSeries("Reservoir","Lake Mendocino",
		"Conservation", "Stor-ZONE")
	curLMStorageCon = LMstorageConTS.getValue(step+1)
	storRatio = curLMStorage/curLMStorageCon
	if storRatio > 0.82:
		YearType = 1
	elif storRatio > 0.64:
		YearType = 2
	else:
		YearType = 3
#May
if mon == 5 and day ==1:
	LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
	curLMStorage = LMstorageTS.getValue(step+1)
	LMstorageConTS = network.getTimeSeries("Reservoir","Lake Mendocino",
		"Conservation", "Stor-ZONE")
	curLMStorageCon = LMstorageConTS.getValue(step+1)
	storRatio = curLMStorage/curLMStorageCon
	if storRatio > 0.625:
		YearType = 1
	elif storRatio > 0.47:
		YearType = 2
	else:
		YearType = 3
#June
if mon == 6 and day ==1:
	LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
	curLMStorage = LMstorageTS.getValue(step+1)
	LMstorageConTS = network.getTimeSeries("Reservoir","Lake Mendocino",
		"Conservation", "Stor-ZONE")
	curLMStorageCon = LMstorageConTS.getValue(step+1)
	storRatio = curLMStorage/curLMStorageCon
	if storRatio > 0.55:
		YearType = 1
	elif storRatio > 0.40:
		YearType = 2
	else:
		YearType = 3

#Save Hydrologic Index
currentVariable.setValue(currentRuntimestep, YearType)
```

A.3 Lake Mendocino Cumulative Inflow Hydrologic Index – Evaluated Weekly

The state variable code (script) for the Lake Mendocino cumulative inflow, evaluated weekly, hydrologic index is shown below. This script is evaluated by HEC-ResSim each time-step during the model simulation. Output from the script is the hydrologic index (referred to as YearType). HEC-ResSim uses the hydrologic index when setting minimum flows from Lake Mendocino and Lake Sonoma. The "clean up" section of the state variable script has been omitted and only the computation section is included below.

```
#####
##### STATE VARIABLE SCRIPT INITIALIZATION SECTION
#####

from hec.script import Constants
from hec.hecmath import TimeSeriesMath, DSS, DSSFile
from hec.script import  ClientAppWrapper

#Load the cumulative inflow thresholds – these are annual pattern time-series
def initStateVariable(currentVariable, network):

        tw=network.getRssRun().getCurrentComputeBlockRunTimeWindow()
        tws = tw.getTimeWindowString()

        global simulation
        module = ClientAppWrapper.getCurrentModule()
        simulation = module.getSimulation()
        simulationDssFilename = simulation.getOutputDSSFilePath()
        dssFile = "shared/ExternalTimeSeries.dss"
        parts = simulationDssFilename.split("/rss")
        directory, directory2 = parts
        dssFileName = '/'.join([directory, dssFile])
        DssFile = DSS.open(dssFileName, tws)

        inflowState3 = DssFile.read("//LAKE MENDOCINO-
                CUMULATIVEINFLOW/INFLOW//1DAY/304/")
        inflowState3TS = inflowState3.getData()
        currentVariable.localTimeSeriesNew("inflowState3", inflowState3TS)

        inflowState2 = DssFile.read("//LAKE MENDOCINO-
                CUMULATIVEINFLOW/INFLOW//1DAY/113/")
        inflowState2TS = inflowState2.getData()
        currentVariable.localTimeSeriesNew("inflowState2", inflowState2TS)

        return Constants.TRUE

#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####

from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
year = curTime.year()
mon = curTime.month()
day = curTime.day()
```

```
dayNum = curTime.dayOfWeek()

#Set simulation time variables
startTime = currentRuntimestep.getRunTimeWindow().getStartTime()
endTime = currentRuntimestep.getRunTimeWindow().getEndTime()
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime)

#Create time variable for beginning of Dry Spring period (June 1)
datestr = "01JUN" + str(year)
bgnDrySpring = HecTime()
bgnDrySpring.setDate(datestr)

#Create time variable beginning of year (Jan 1)
datestr = "01JAN" + str(year)
bgnYear = HecTime()
bgnYear.setDate(datestr)

#Create Lake Mendocino Inflow Time Series object
LMinflowTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Flow-IN")

#------------------------------Calculate Cumulative Flow----------------------------------
#Get previous value from slave variable
LMCummInflowSV = network.getStateVariable("slave_LMCummInflow")
LMCummInflowTS = LMCummInflowSV.getTimeSeries()

#Get previous cumulative Lake Mendocino Inflow
if mon == 10 and day == 1:
        prevLMCummInflow = 0
else:
        prevLMCummInflow = LMCummInflowTS.getPreviousValue(currentRuntimestep)

#Convert to acre-feet/day
curLMInflow = LMinflowTS.getValue(step+1)*1.9835

#Calculate Cumulative Inflow
curLMCummInflow = prevLMCummInflow + curLMInflow
LMCummInflowSV.setValue(currentRuntimestep, curLMCummInflow)
#----------------------------------------------------------------------------------------

#Initialize YearType (Hydrologic Index)
YearType = currentVariable.getPreviousValue(currentRuntimestep)

#Get cumulative inflow thresholds – from initialization
state3TS = currentVariable.localTimeSeriesGet("inflowState3")
state2TS = currentVariable.localTimeSeriesGet("inflowState2")
state3 = state3TS.getValue(curTime)
state2 = state2TS.getValue(curTime)

#Compute hydrologic index by comparing cumulative Lake Mendocino inflow to
#thresholds (dayNum = 2 → index is evaluated every Monday)
if curTime.julian()>=bgnYear.julian() and curTime.julian()<bgnDrySpring.julian():
        if dayNum == 2:
                if curLMCummInflow < state3:
                        YearType = 3
```

```
        elif curLMCummInflow < state2:
            YearType = 2
        else:
            YearType = 1
```

#Save Hydrologic Index

```
currentVariable.setValue(currentRuntimestep, YearType)
```

A.4 Lake Mendocino Storage Hydrologic Index – Evaluated Weekly

The state variable code (script) for the Lake Mendocino storage, evaluated weekly, hydrologic index is shown below. This script is evaluated by HEC-ResSim each time-step during the model simulation. Output from the script is the hydrologic index (referred to as YearType). HEC-ResSim uses the hydrologic index when setting minimum flows from Lake Mendocino and Lake Sonoma. The "clean up" section of the state variable script has been omitted and only the computation section is included below.

```
#####
##### STATE VARIABLE SCRIPT INITIALIZATION SECTION
#####

from hec.script import Constants
from hec.hecmath import TimeSeriesMath, DSS, DSSFile
from hec.script import  ClientAppWrapper

#Load the storage thresholds – these are annual pattern time-series
def initStateVariable(currentVariable, network):

    tw=network.getRssRun().getCurrentComputeBlockRunTimeWindow()
    tws = tw.getTimeWindowString()

    global simulation
    module = ClientAppWrapper.getCurrentModule()
    simulation = module.getSimulation()
    simulationDssFilename = simulation.getOutputDSSFilePath()
    dssFile = "shared/ExternalTimeSeries.dss"
    parts = simulationDssFilename.split("/rss")
    directory, directory2 = parts
    dssFileName = '/'.join([directory, dssFile])
    DssFile = DSS.open(dssFileName, tws)

    gcCrit = DssFile.read("//LAKE MENDOCINO-STORAGECURVE/STOR-
        ZONE//1DAY/304/")
    gcCritTS = gcCrit.getData()
    currentVariable.localTimeSeriesNew("gcCrit", gcCritTS)

    gcDry = DssFile.read("//LAKE MENDOCINO-STORAGECURVE/STOR-
        ZONE//1DAY/113/")
    gcDryTS = gcDry.getData()
    currentVariable.localTimeSeriesNew("gcDry", gcDryTS)

    return Constants.TRUE
```

```
#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####

from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
year = curTime.year()
mon = curTime.month()
day = curTime.day()
dayNum = curTime.dayOfWeek()

#Set simulation time variables
startTime = currentRuntimestep.getRunTimeWindow().getStartTime()
endTime = currentRuntimestep.getRunTimeWindow().getEndTime()
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime)

#Get storage thresholds – from initialization
LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
curLMStorage = LMstorageTS.getValue(step+1)
gc1TS = currentVariable.localTimeSeriesGet("gcCrit")
gc1 = gc1TS.getValue(curTime)
gc2TS = currentVariable.localTimeSeriesGet("gcDry")
gc2 = gc2TS.getValue(curTime)

#Initialize YearType (Hydrologic Index)
YearType = currentVariable.getPreviousValue(currentRuntimestep)

#Compute hydrologic index by comparing current Lake Mendocino storage to
#thresholds (dayNum = 2 → index is evaluated every Monday)
if dayNum == 2:
	if curLMStorage < gc1:
		YearType = 3
	elif curLMStorage < gc2:
		YearType = 2
	else:
		YearType = 1
#Save Hydrologic Index
currentVariable.setValue(currentRuntimestep, YearType)
```

A.5 Dual Index – Cumulative Lake Mendocino Inflow and Lake Sonoma Storage Hydrologic Index

Two state variable scripts were used in the dual index, cumulative Lake Mendocino inflow and Lake Sonoma storage hydrologic index. The state variable code (script) for the Lake Sonoma storage hydrologic index is shown below; the cumulative Lake Mendocino Inflow index is shown in Section A.1. The scripts are evaluated by HEC-ResSim each time-step during the model simulation. Outputs from the scripts are the hydrologic index. HEC-ResSim uses the hydrologic index computed from the Lake Sonoma storage hydrologic index when setting minimum flows from Lake Sonoma (the cumulative Lake Mendocino inflow index was used to set minimum flows from Lake Mendocino). The "clean up" section of the state variable script has been omitted and only the computation section is included below.

```
#####
##### STATE VARIABLE SCRIPT INITIALIZATION SECTION
#####

from hec.script import Constants
from hec.hecmath import TimeSeriesMath, DSS, DSSFile
from hec.script import  ClientAppWrapper

#Load the storage thresholds – these are annual pattern time-series
def initStateVariable(currentVariable, network):

	tw=network.getRssRun().getCurrentComputeBlockRunTimeWindow()
	tws = tw.getTimeWindowString()

	global simulation
	module = ClientAppWrapper.getCurrentModule()
	simulation = module.getSimulation()
	simulationDssFilename = simulation.getOutputDSSFilePath()
	dssFile = "shared/ExternalTimeSeries.dss"
	parts = simulationDssFilename.split("/rss")
	directory, directory2 = parts
	dssFileName = '/'.join([directory, dssFile])
	DssFile = DSS.open(dssFileName, tws)

	gcCrit = DssFile.read("//LAKE SONOMA-STORAGECURVE/STOR-ZONE//1DAY/310/")
	gcCritTS = gcCrit.getData()
	currentVariable.localTimeSeriesNew("gcCrit", gcCritTS)

	gcDry = DssFile.read("//LAKE SONOMA-STORAGECURVE/STOR-ZONE//1DAY/113/")
	gcDryTS = gcDry.getData()
	currentVariable.localTimeSeriesNew("gcDry", gcDryTS)

	return Constants.TRUE

#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####

from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
year = curTime.year()
mon = curTime.month()
day = curTime.day()
dayNum = curTime.dayOfWeek()
```

#Set simulation time variables
startTime = currentRuntimestep.getRunTimeWindow().getStartTime()
endTime = currentRuntimestep.getRunTimeWindow().getEndTime()
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime)

#Get storage thresholds – from initialization
LSstorageTS = network.getTimeSeries("Reservoir","Lake Sonoma", "Pool", "Stor")
curLSStorage = LSstorageTS.getValue(step+1)
gc1TS = currentVariable.localTimeSeriesGet("gcCrit")
gc1 = gc1TS.getValue(curTime)
gc2TS = currentVariable.localTimeSeriesGet("gcDry")
gc2 = gc2TS.getValue(curTime)

#Initialize YearTypeS (Hydrologic Index)
YearTypeS = currentVariable.getPreviousValue(currentRuntimestep)

#Compute hydrologic index by comparing current Lake Sonoma storage to
#thresholds (dayNum = 2 ➔ index is evaluated every Monday)
```
if dayNum == 2:
	if curLSStorage < gc1:
		YearTypeS = 3
	elif curLSStorage < gc2:
		YearTypeS = 2
	else:
		YearTypeS = 1
```

#Save Hydrologic Index
currentVariable.setValue(currentRuntimestep, YearTypeS)

A.6 Water Balance Hydrologic Index

The state variable code (script) for the Lake Mendocino water balance hydrologic index is shown below. This script is evaluated by HEC-ResSim each time-step during the model simulation. Output from the script is the hydrologic index (referred to as hydroIndex). HEC-ResSim uses the hydrologic index when setting minimum flows from Lake Mendocino and Lake Sonoma. The "clean up" section of the state variable script has been omitted and only the computation section is included below.

#####
STATE VARIABLE SCRIPT INITIALIZATION SECTION
#####

from hec.script import Constants
from hec.hecmath import TimeSeriesMath, DSS, DSSFile
from hec.script import ClientAppWrapper

#Load the meteorologic forecast – this time-series is used by the script for setting
#the meteorologic forecasts: Normal, Dry, or Critical. This was computed
#by evaluating the historic Lake Mendocino inflow dataset.
```
def initStateVariable(currentVariable, network):

	tw=network.getRssRun().getCurrentComputeBlockRunTimeWindow()
	tws = tw.getTimeWindowString()
```

```
        global simulation
        module = ClientAppWrapper.getCurrentModule()
        simulation = module.getSimulation()
        simulationDssFilename = simulation.getOutputDSSFilePath()
        dssFile = "shared/ExternalTimeSeries.dss"
        parts = simulationDssFilename.split("/rss")
        directory, directory2 = parts
        dssFileName = '/'.join([directory, dssFile])
        DssFile = DSS.open(dssFileName, tws)

        forecast = DssFile.read("//FORCASTEDCONDITION/FORECAST//1MON/PREVIOUS
                MONTH/")
        forecastTS = forecast.getData()
        currentVariable.localTimeSeriesNew("forecast", forecastTS)

        return Constants.TRUE

#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####

from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS

#The following variables are used by the water balance, monthly inflows
#(percentiles from the Lake Mendocino dataset), monthly loss, monthly Lake
#Mendocino evaporation, end and beginning of month Lake Mendocino
#conservation storage, and minimum flows
monthList = ("JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC")
monthInflow5P = (8813, 9856, 3742, 2772, 3515, 2801, 2370, 2391, 3262, 2419, 3401,
                3309)
monthInflow10P = (12508, 11372, 5447, 2937, 3707, 2824, 2372, 2421, 4255, 2508,
                5633, 5288)
monthInflow25P = (20111, 20403, 11129, 3441, 5500, 5777, 5445, 5465, 4485, 7535,
                8311, 12736)
monthInflow50P = (35912, 29854, 17321, 4714, 5561, 5785, 5445, 5465, 6537, 7921,
                9871, 27191)
monthLoss = (504, 452, 944, 1161, 1438, 4068, 5467, 5155, 4903, 4170, 627, 526)
monthEvap = (114, 118, 203, 321, 502, 722, 883, 816, 755, 488, 214, 121)
beginMonthStorageGC = (68409, 68409, 68409, 86400, 105930, 110967, 110967, 110967,
                110967, 110967, 68409, 68409)
endMonthStorageGC = (68409, 68409, 86400, 105930, 110967, 110967, 110967, 110967,
                110967, 68409, 68409, 68409)
minFlowNormalInc = (9223, 8330, 9224, 11008, 11375, 7438, 7686, 7686, 7438, 7438,
                7438, 7438)
minFlowDryInc = (4612, 4165, 4612, 4463, 4611, 4463, 4612, 4611, 4463, 4463, 4463,
                4463)
minFlowCriticalInc = (1537, 1388, 1538, 1487, 1537, 1488, 1537, 1537, 1488, 1488, 1488,
                1488)
minFlowNormalFut = (79408, 70185, 61855, 52631, 41623, 30248, 22810, 15124, 7438, 0,
                0, 0)
```

```
minFlowDryFut = (40612, 36000, 31835, 27223, 22760, 18149, 13686, 9074, 4463, 0, 0,
        0)
minFlowCriticalFut = (13537, 12000, 10612, 9074, 7587, 6050, 4562, 3025, 1488, 0, 0, 0)
monthDemandFut = (24092, 23588, 23136, 22192, 21031, 19593, 15526, 10058, 4903, 0,
        0, 0)
monthEvapFut = (4434, 4320, 4202, 400, 3679, 3176, 2454, 1571, 755, 0, 0, 0)

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
year = curTime.year()
mon = curTime.month()
day = curTime.day()
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime)

#Initialize hydroIndex (Hydrologic Index)
hydroIndex = currentVariable.getPreviousValue(currentRuntimestep)

#bufferVal = buffer used by the water balance
#autoDryIndex = automatically set hydrologic index to dry if Lake Mendocino
#Storage drops below this value
#autoCriticalIndex = automatically set hydrologic index to critical if Lake
#Mendocino Storage drops below this value
#delay = if delay = 0 then no delay, if delay = 1 then delay changing the
#hydrologic index
bufferVal = 30000
autoDryIndex = 30000
autoCriticalIndex = 20000
delay = 1
#Get the current Lake Mendocino Storage
LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
curLMStorage = LMstorageTS.getValue(step+1)

#Below is the logic for the water balance-----------------------------------------------
j = mon
#The water balance is only evaluated between Jan and Sep on the first of the
#month
if mon < 10 and day == 1:
    endOfMonthStorage = 0
    waterStore = 0
    monthUsable = 0
    totalUsable = 0
    lastIndex = 1
    lastIndex2 = 1
    #Get the forecast
    forecastTS = currentVariable.localTimeSeriesGet("forecast")
    forecastedIndex = forecastTS.getValue(curTime)
    endofMonthStorageDummy = 0
    if forecastedIndex == 3: #Normal Forecast
        i = (j - 1)
        while i < 9:
            #starting with the current month, loop through months until
            #September and compute the end of month storage, storable
            #water, and total usable water (for the remainder of the water
            #year)
```

```
                if j - 1 > i:
                        endOfMonthStorage = 0
                elif j - 1 < i:
                        endOfMonthStorage = endOfMonthStorage +
                        monthInflow50P[i] - monthDemand[i] - monthEvap[i] -
                        minFlowNormalInc[i]
                else:
                        endOfMonthStorage = curLMStorage + monthInflow50P[i] -
                        monthDemand[i] - monthEvap[i] - minFlowNormalInc[i]

                if endOfMonthStorage < 0:
                        endOfMonthStorage = 0

                if endOfMonthStorage > endMonthStorageGC[i]:
                        endOfMonthStorage = endMonthStorageGC[i]

                if endofMonthStorageDummy == 0:
                        waterStore = endOfMonthStorage - curLMStorage
                        endofMonthStorageDummy = endOfMonthStorage
                else:
                        waterStore = endOfMonthStorage - endofMonthStorageDummy
                        endofMonthStorageDummy = endOfMonthStorage

                if waterStore < 0:
                        waterStore = 0

                monthUsable =  waterStore + monthDemand[i] + monthEvap[i] +
                minFlowNormalInc[i]
                if monthUsable > monthInflow50P[i]:
                        monthUsable = monthInflow50P[i]

                totalUsable += monthUsable

                i += 1

        #Using the total usable water, compute the water available for
        #meeting minimum flows
        availableWater = totalUsable + curLMStorage - bufferVal
        futureDemands = monthDemandFut[mon - 1] + monthEvapFut[mon - 1]
        availableMinusDemands = availableWater - futureDemands
        if (availableMinusDemands < 0):
                availableMinusDemands = 0

elif forecastedIndex == 2: #Dry Forecast
        i = (j - 1)
        while i < 9:
                #starting with the current month, loop through months until
                #September and compute the end of month storage, storable
                #water, and total usable water (for the remainder of the water
                #year)
                if j - 1 > i:
                        endOfMonthStorage = 0
                elif j - 1 < i:
```

```
            endOfMonthStorage = endOfMonthStorage +
            monthInflow25P[i] - monthDemand[i] - monthEvap[i] -
            minFlowDryInc[i]
        else:
            endOfMonthStorage = curLMStorage + monthInflow25P[i] -
            monthDemand[i] - monthEvap[i] - minFlowDryInc[i]

        if endOfMonthStorage < 0:
            endOfMonthStorage = 0

        if endOfMonthStorage > endMonthStorageGC[i]:
            endOfMonthStorage = endMonthStorageGC[i]

        if endofMonthStorageDummy == 0:
            waterStore = endOfMonthStorage - curLMStorage
            endofMonthStorageDummy = endOfMonthStorage
        else:
            waterStore = endOfMonthStorage - endofMonthStorageDummy
            endofMonthStorageDummy = endOfMonthStorage

        if waterStore < 0:
            waterStore = 0

        monthUsable = waterStore + monthDemand[i] + monthEvap[i] +
        minFlowDryInc[i]
        if monthUsable > monthInflow25P[i]:
            monthUsable = monthInflow25P[i]

        totalUsable += monthUsable

        i += 1
    #Using the total usable water, compute the water available for
    #meeting minimum flows
    availableWater = totalUsable + curLMStorage - bufferVal
    futureDemands = monthDemandFut[mon - 1] + monthEvapFut[mon - 1]
    availableMinusDemands = availableWater - futureDemands
    if (availableMinusDemands < 0):
        availableMinusDemands = 0

else: #Critical Forecast
    i = (j - 1)
    while i < 9:
        #starting with the current month, loop through months until
        #September and compute the end of month storage, storable
        #water, and total usable water (for the remainder of the water
        #year)
        if j - 1 > i:
            endOfMonthStorage = 0
        elif j - 1 < i:
            endOfMonthStorage = endOfMonthStorage + monthInflow5P[i]
            - monthDemand[i] - monthEvap[i] - minFlowCriticalInc[i]
        else:
            endOfMonthStorage = curLMStorage + monthInflow5P[i] -
            monthDemand[i] - monthEvap[i] - minFlowCriticalInc[i]
```

```
			if endOfMonthStorage < 0:
				endOfMonthStorage = 0

			if endOfMonthStorage > endMonthStorageGC[i]:
				endOfMonthStorage = endMonthStorageGC[i]

			if endofMonthStorageDummy == 0:
				waterStore = endOfMonthStorage - curLMStorage
				endofMonthStorageDummy = endOfMonthStorage
			else:
				waterStore = endOfMonthStorage - endofMonthStorageDummy
				endofMonthStorageDummy = endOfMonthStorage

			if waterStore < 0:
				waterStore = 0

			monthUsable = waterStore + monthDemand[i] + monthEvap[i] +
			minFlowCriticalInc[i]
			if monthUsable > monthInflow5P[i]:
				monthUsable = monthInflow5P[i]

			totalUsable += monthUsable

			i += 1

		#Using the total usable water, compute the water available for
		#meeting minimum flows
		availableWater = totalUsable + curLMStorage - bufferVal
		futureDemands = monthDemandFut[mon - 1] + monthEvapFut[mon - 1]
		availableMinusDemands = availableWater - futureDemands
		if (availableMinusDemands < 0):
			availableMinusDemands = 0

	#Compare the available water to minimum flows to compute the hydrologic
	#index
	if (availableMinusDemands > minFlowNormalFut[mon - 1]):
		hydroIndex = 1
	elif (availableMinusDemands > minFlowDryFut[mon - 1]):
		hydroIndex = 2
	else:
		hydroIndex = 3

#Use the auto-index option to set the hydrologic index (based on current storage)
if day == 1:
	if curLMStorage < autoCriticalIndex:
		hydroIndex = 3
	elif hydroIndex == 1 and curLMStorage < autoDryIndex:
		hydroIndex = 2

	hydroIndexSV = network.getStateVariable("Slave_hydroIndex_prevVal")
	hydroIndexSV.setValue(currentRuntimestep, hydroIndex)
```

```
        #Option for delaying changing the hydrologic index
        if delay == 1:
                daysInMonth = [0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
                if curTime.isLeap(year):
                        daysInMonth[2] = 29
                else:
                        daysInMonth[2] = 28
                curDaysInMonth = daysInMonth[mon]
                prevMonth = mon - 1
                if prevMonth == 0:
                        prevMonth = 12
                prev2Month = prevMonth - 1
                if prev2Month == 0:
                        prev2Month = 12
                prevDaysInMonth = daysInMonth[prevMonth]
                prev2DaysInMonth = daysInMonth[prev2Month]
                curvalTS = hydroIndexSV.getTimeSeries()
                lastIndex = curvalTS.getLaggedValue(currentRuntimestep, prevDaysInMonth)
                lastIndex2 = curvalTS.getLaggedValue(currentRuntimestep, prevDaysInMonth
                + prev2DaysInMonth)
                if (hydroIndex > lastIndex):
                        hydroIndex =  hydroIndex
                elif (lastIndex == hydroIndex):
                        hydroIndex = hydroIndex
                elif (lastIndex == lastIndex2):
                        hydroIndex = lastIndex
                else:
                        hydroIndex = lastIndex

#Save the hydrologic index
currentVariable.setValue(currentRuntimestep, hydroIndex)
```

A.7 Cumulative Inflow plus Water Balance Hydrologic Index

The state variable code (script) for the Lake Mendocino cumulative inflow plus water balance hydrologic index is shown below. This script is evaluated by HEC-ResSim each time-step during the model simulation. Output from the script is the hydrologic index (referred to as hydroIndex). HEC-ResSim uses the hydrologic index when setting minimum flows from Lake Mendocino and Lake Sonoma. The "clean up" section of the state variable script has been omitted and only the computation section is included below.

```
#####
##### STATE VARIABLE SCRIPT INITIALIZATION SECTION
#####

from hec.script import Constants
from hec.hecmath import TimeSeriesMath, DSS, DSSFile
from hec.script import  ClientAppWrapper

#Load the meteorologic forecast – this time-series is used by the script for setting
#the meteorologic forecasts: Normal, Dry, or Critical.  This was computed
#by evaluating the historic Lake Mendocino inflow dataset.
def initStateVariable(currentVariable, network):
```

```
        tw=network.getRssRun().getCurrentComputeBlockRunTimeWindow()
        tws = tw.getTimeWindowString()

        global simulation
        module = ClientAppWrapper.getCurrentModule()
        simulation = module.getSimulation()
        simulationDssFilename = simulation.getOutputDSSFilePath()
        dssFile = "shared/ExternalTimeSeries.dss"
        parts = simulationDssFilename.split("/rss")
        directory, directory2 = parts
        dssFileName = '/'.join([directory, dssFile])
        DssFile = DSS.open(dssFileName, tws)

        forecast = DssFile.read("//FORCASTEDCONDITION/FORECAST//1MON/PREVIOUS
                MONTH/")
        forecastTS = forecast.getData()
        currentVariable.localTimeSeriesNew("forecast", forecastTS)

        return Constants.TRUE



#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####

from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS

#The following variables are used by the water balance, monthly inflows
#(percentiles from the Lake Mendocino dataset), monthly loss, monthly Lake
#Mendocino evaporation, end and beginning of month Lake Mendocino
#conservation storage, and minimum flows
monthList = ("JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC")
monthInflow5P = (8813, 9856, 3742, 2772, 3515, 2801, 2370, 2391, 3262, 2419, 3401,
                3309)
monthInflow10P = (12508, 11372, 5447, 2937, 3707, 2824, 2372, 2421, 4255, 2508,
                5633, 5288)
monthInflow25P = (20111, 20403, 11129, 3441, 5500, 5777, 5445, 5465, 4485, 7535,
                8311, 12736)
monthInflow50P = (35912, 29854, 17321, 4714, 5561, 5785, 5445, 5465, 6537, 7921,
                9871, 27191)
monthLoss = (504, 452, 944, 1161, 1438, 4068, 5467, 5155, 4903, 4170, 627, 526)
monthEvap = (114, 118, 203, 321, 502, 722, 883, 816, 755, 488, 214, 121)
beginMonthStorageGC = (68409, 68409, 68409, 86400, 105930, 110967, 110967, 110967,
                110967, 110967, 68409, 68409)
endMonthStorageGC = (68409, 68409, 86400, 105930, 110967, 110967, 110967, 110967,
                110967, 68409, 68409, 68409)
minFlowNormalInc = (9223, 8330, 9224, 11008, 11375, 7438, 7686, 7686, 7438, 7438,
                7438, 7438)
minFlowDryInc = (4612, 4165, 4612, 4463, 4611, 4463, 4612, 4611, 4463, 4463, 4463,
                4463)
```

```
minFlowCriticalInc = (1537, 1388, 1538, 1487, 1537, 1488, 1537, 1537, 1488, 1488, 1488,
        1488)
minFlowNormalFut = (79408, 70185, 61855, 52631, 41623, 30248, 22810, 15124, 7438, 0,
        0, 0)
minFlowDryFut = (40612, 36000, 31835, 27223, 22760, 18149, 13686, 9074, 4463, 0, 0,
        0)
minFlowCriticalFut = (13537, 12000, 10612, 9074, 7587, 6050, 4562, 3025, 1488, 0, 0, 0)
monthDemandFut = (24092, 23588, 23136, 22192, 21031, 19593, 15526, 10058, 4903, 0,
        0, 0)
monthEvapFut = (4434, 4320, 4202, 400, 3679, 3176, 2454, 1571, 755, 0, 0, 0)

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
year = curTime.year()
mon = curTime.month()
day = curTime.day()
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime)

#Set simulation start time variables
startTime = currentRuntimestep.getRunTimeWindow().getStartTime()
endTime = currentRuntimestep.getRunTimeWindow().getEndTime()
styear = startTime.year()
stmon = startTime.month()
stday = startTime.day()

#Set date for the beginning of the Water Year
if mon >= 10:
        datestr = "1OCT" + str(year)
else:
        datestr = "1OCT" + str(year-1)

#Define starting date of the water year in order to accumulate Lake Mendocino
#Inflow
bgnWY = HecTime()
bgnWY.setDate(datestr)
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime)
CummPeriod = curTime.julian() - bgnWY.julian() + 1

#Create Lake Mendocino Inflow Time Series object
LMinflowTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Flow-IN")

#------------------------------Calculate Cumulative Flow---------------------------------
#Get previous value from slave variable
LMCummInflowSV = network.getStateVariable("slave_LMCummInflow")
LMCummInflowTS = LMCummInflowSV.getTimeSeries()

#Get previous cumulative LM Inflow
if mon == 10 and day ==1:
        prevLMCummInflow = 0
else:
        prevLMCummInflow = LMCummInflowTS.getPreviousValue(currentRuntimestep)

#Convert to acre-feet/day
curLMInflow = LMinflowTS.getValue(step+1)*1.9835
```

```
#Calculate Cumulative Inflow
curLMCummInflow = prevLMCummInflow + curLMInflow
LMCummInflowSV.setValue(currentRuntimestep, curLMCummInflow)
#------------------------------------------------------------------------------------------

#Initialize hydrologic index (YearType)
YearType = currentVariable.getPreviousValue(currentRuntimestep)

#bufferVal = buffer used by the water balance
#autoDryIndex = automatically set hydrologic index to dry if Lake Mendocino
#Storage drops below this value
#autoCriticalIndex = automatically set hydrologic index to critical if Lake
#Mendocino Storage drops below this value
bufferVal = 30000
autoDryIndex = 30000
autoCriticalIndex = 20000

#Get the current Lake Mendocino storage
LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
curLMStorage = LMstorageTS.getValue(step+1)
#Evaluate Lake Mendocino Cumulative Inflow (compare to thresholds) to set the
#Hydrologic Index (first of the month)
#January
if mon == 1 and day == 1:
	if curLMCummInflow < 19250:
		YearType = 3
	elif curLMCummInflow < 25500 and curLMCummInflow >= 19250:
		YearType = 2
	else:
		YearType = 1
#February
elif mon == 2 and day == 1:
	if curLMCummInflow < 37000:
		YearType = 3
	elif curLMCummInflow < 54000 and curLMCummInflow >= 37000:
		YearType = 2
	else:
		YearType = 1
#March
elif mon == 3 and day == 1:
	if curLMCummInflow < 49000:
		YearType = 3
	elif curLMCummInflow < 72000 and curLMCummInflow >= 49000:
		YearType = 2
	else:
		YearType = 1
#April
elif mon == 4 and day == 1:
	if curLMCummInflow < 54000:
		YearType = 3
	elif curLMCummInflow < 93500 and curLMCummInflow >= 54000:
		YearType = 2
	else:
```

```
        YearType = 1
#May
elif mon == 5 and day == 1:
    if curLMCummInflow < 54000:
        YearType = 3
    elif curLMCummInflow < 99000 and curLMCummInflow >= 54000:
        YearType = 2
    else:
        YearType = 1

#Use the water balance approach from June 1 - Dec 1
elif mon < 10 and day == 1:
    j = mon
    if day == 1:
        endOfMonthStorage = 0
        waterStore = 0
        monthUsable = 0
        totalUsable = 0
        lastIndex = 1
        lastIndex2 = 1
        #Get the forecast
        forecastTS = currentVariable.localTimeSeriesGet("forecast")
        forecastedIndex = 3
        endofMonthStorageDummy = 0
        if forecastedIndex == 3: #normal forecast
            i = (j - 1)
            while i < 9:
                #starting with the current month, loop through months
                #until September and compute the end of month
                #storage, storable water, and total usable water (for
                #the remainder of the water year)
                if j - 1 > i:
                    endOfMonthStorage = 0
                elif j - 1 < i:
                    endOfMonthStorage = endOfMonthStorage +
                    monthInflow50P[i] - monthDemand[i] - monthEvap[i] -
                    minFlowNormalInc[i]
                else:
                    endOfMonthStorage = curLMStorage +
                    monthInflow50P[i] - monthDemand[i] - monthEvap[i] -
                    minFlowNormalInc[i]

                if endOfMonthStorage < 0:
                    endOfMonthStorage = 0

                if endOfMonthStorage > endMonthStorageGC[i]:
                    endOfMonthStorage = endMonthStorageGC[i]

                if endofMonthStorageDummy == 0:
                    waterStore = endOfMonthStorage - curLMStorage
                    endofMonthStorageDummy = endOfMonthStorage
                else:
                    waterStore = endOfMonthStorage -
                    endofMonthStorageDummy
```

```
                endofMonthStorageDummy = endOfMonthStorage

            if waterStore < 0:
                waterStore = 0

            monthUsable =  waterStore + monthDemand[i] +
            monthEvap[i] + minFlowNormalInc[i]
            if monthUsable > monthInflow50P[i]:
                monthUsable = monthInflow50P[i]

            totalUsable += monthUsable

            i += 1

        #Using the total usable water, compute the water available for
        #meeting minimum flows
        availableWater = totalUsable + curLMStorage - bufferVal
        futureDemands = monthDemandFut[mon - 1] + monthEvapFut[mon -
        1]
        availableMinusDemands = availableWater - futureDemands
        if (availableMinusDemands < 0):
            availableMinusDemands = 0

    elif forecastedIndex == 2: #dry forecast
        i = (j - 1)
        while i < 9:
            #starting with the current month, loop through months
            #until September and compute the end of month
            #storage, storable water, and total usable water (for
            #the remainder of the water year)
            if j - 1 > i:
                endOfMonthStorage = 0
            elif j - 1 < i:
                endOfMonthStorage = endOfMonthStorage +
                monthInflow25P[i] - monthDemand[i] - monthEvap[i] -
                minFlowDryInc[i]
            else:
                endOfMonthStorage = curLMStorage +
                monthInflow25P[i] - monthDemand[i] - monthEvap[i] -
                minFlowDryInc[i]

            if endOfMonthStorage < 0:
                endOfMonthStorage = 0

            if endOfMonthStorage > endMonthStorageGC[i]:
                endOfMonthStorage = endMonthStorageGC[i]

            if endofMonthStorageDummy == 0:
                waterStore = endOfMonthStorage - curLMStorage
                endofMonthStorageDummy = endOfMonthStorage
            else:
                waterStore = endOfMonthStorage -
                endofMonthStorageDummy
                endofMonthStorageDummy = endOfMonthStorage
```

```
                if waterStore < 0:
                        waterStore = 0

                monthUsable = waterStore + monthDemand[i] + monthEvap[i]
                + minFlowDryInc[i]
                if monthUsable > monthInflow25P[i]:
                        monthUsable = monthInflow25P[i]

                totalUsable += monthUsable

                i += 1

        #Using the total usable water, compute the water available for
        #meeting minimum flows
        availableWater = totalUsable + curLMStorage - bufferVal
        futureDemands = monthDemandFut[mon - 1] + monthEvapFut[mon -
        1]
        availableMinusDemands = availableWater - futureDemands
        if (availableMinusDemands < 0):
                availableMinusDemands = 0

else: #critical forecast
        i = (j - 1)
        while i < 9:
                #starting with the current month, loop through months
                #until September and compute the end of month
                #storage, storable water, and total usable water (for
                #the remainder of the water year)
                if j - 1 > i:
                        endOfMonthStorage = 0
                elif j - 1 < i:
                        endOfMonthStorage = endOfMonthStorage +
                        monthInflow5P[i] - monthDemand[i] - monthEvap[i] -
                        minFlowCriticalInc[i]
                else:
                        endOfMonthStorage = curLMStorage + monthInflow5P[i]
                        - monthDemand[i] - monthEvap[i] -
                        minFlowCriticalInc[i]

                if endOfMonthStorage < 0:
                        endOfMonthStorage = 0

                if endOfMonthStorage > endMonthStorageGC[i]:
                        endOfMonthStorage = endMonthStorageGC[i]

                if endofMonthStorageDummy == 0:
                        waterStore = endOfMonthStorage - curLMStorage
                        endofMonthStorageDummy = endOfMonthStorage
                else:
                        waterStore = endOfMonthStorage -
                        endofMonthStorageDummy
                        endofMonthStorageDummy = endOfMonthStorage
```

```
                if waterStore < 0:
                    waterStore = 0

                monthUsable = waterStore + monthDemand[i] + monthEvap[i]
                + minFlowCriticalInc[i]
                if monthUsable > monthInflow5P[i]:
                    monthUsable = monthInflow5P[i]

                totalUsable += monthUsable

                i += 1

            #Using the total usable water, compute the water available for
            #meeting minimum flows
            availableWater = totalUsable + curLMStorage - bufferVal
            futureDemands = monthDemandFut[mon - 1] + monthEvapFut[mon -
            1]
            availableMinusDemands = availableWater - futureDemands
            if (availableMinusDemands < 0):
                availableMinusDemands = 0

        #Compare the available water to minimum flows to compute the
        #hydrologic index
        if (availableMinusDemands > minFlowNormalFut[mon - 1]):
            YearType = 1
        elif (availableMinusDemands > minFlowDryFut[mon - 1]):
            YearType = 2
        else:
            YearType = 3

#Use the auto-index option to set the hydrologic index (based on current storage)
if day == 1:
    if curLMStorage < autoCriticalIndex:
        YearType = 3
    elif YearType == 1 and curLMStorage < autoDryIndex:
        YearType = 2

#Save the hydrologic index
currentVariable.setValue(currentRuntimestep, YearType)
```

A.8 Water Balance with Ensemble Hydrologic Index

The state variable code (script) for the Lake Mendocino water balance with ensemble hydrologic index is shown below. This script is evaluated by HEC-ResSim each time-step during the model simulation. Output from the script is the hydrologic index (referred to as hydroIndex). HEC-ResSim uses the hydrologic index when setting minimum flows from Lake Mendocino and Lake Sonoma. The "clean up" section of the state variable script has been omitted and only the computation section is included below.

```
#####
##### STATE VARIABLE SCRIPT INITIALIZATION SECTION
#####

from hec.script import Constants
```

```
from hec.hecmath import TimeSeriesMath, DSS, DSSFile
from hec.script import  ClientAppWrapper

#Load the meteorologic forecast – this time-series is used by the script for setting
#the meteorologic forecasts: Normal, Dry, or Critical.  This was computed
#by evaluating the historic Lake Mendocino inflow dataset.  Also, load the
#ensemble of monthly Lake Mendocino Inflow (98-yearof record)
def initStateVariable(currentVariable, network):

	global simulation
	module = ClientAppWrapper.getCurrentModule()
	simulation = module.getSimulation()
	simulationDssFilename = simulation.getOutputDSSFilePath()
	dssFile = "shared/ExternalTimeSeries.dss"
	parts = simulationDssFilename.split("/rss")
	directory, directory2 = parts
	dssFileName = '/'.join([directory, dssFile])
	DssFile = DSS.open(dssFileName, "01JAN1910 2400 31DEC2008 2400")

	forecast = DssFile.read("//FORCASTEDCONDITION/FORECAST//1MON/PREVIOUS
		MONTH/")
	forecastTS = forecast.getData()
	currentVariable.localTimeSeriesNew("forecast", forecastTS)

	ensemble = DssFile.read("/LM/LAKE MEND INFLOW(CURPVP)/FLOWACRE-
		FEET//1MON/USGS4.2B/")
	ensembleTS = ensemble.getData()
	currentVariable.localTimeSeriesNew("ensemble", ensembleTS)

	return Constants.TRUE

#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####

from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS

#The following variables are used by the water balance, monthly inflows
#(percentiles from the Lake Mendocino dataset), monthly loss, monthly Lake
#Mendocino evaporation, end and beginning of month Lake Mendocino
#conservation storage, and minimum flows
monthList = ("JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC")
monthInflow5P = (8813, 9856, 3742, 2772, 3515, 2801, 2370, 2391, 3262, 2419, 3401,
		3309)
monthInflow10P = (12508, 11372, 5447, 2937, 3707, 2824, 2372, 2421, 4255, 2508,
		5633, 5288)
monthInflow25P = (20111, 20403, 11129, 3441, 5500, 5777, 5445, 5465, 4485, 7535,
		8311, 12736)
monthInflow50P = (35912, 29854, 17321, 4714, 5561, 5785, 5445, 5465, 6537, 7921,
		9871, 27191)
monthLoss = (504, 452, 944, 1161, 1438, 4068, 5467, 5155, 4903, 4170, 627, 526)
monthEvap = (114, 118, 203, 321, 502, 722, 883, 816, 755, 488, 214, 121)
```

```
beginMonthStorageGC = (68409, 68409, 68409, 86400, 105930, 110967, 110967, 110967,
                    110967, 110967, 68409, 68409)
endMonthStorageGC = (68409, 68409, 86400, 105930, 110967, 110967, 110967, 110967,
                    110967, 68409, 68409, 68409)
minFlowNormalInc = (9223, 8330, 9224, 11008, 11375, 7438, 7686, 7686, 7438, 7438,
                    7438, 7438)
minFlowDryInc = (4612, 4165, 4612, 4463, 4611, 4463, 4612, 4611, 4463, 4463, 4463,
                    4463)
minFlowCriticalInc = (1537, 1388, 1538, 1487, 1537, 1488, 1537, 1537, 1488, 1488, 1488,
                    1488)
minFlowNormalFut = (79408, 70185, 61855, 52631, 41623, 30248, 22810, 15124, 7438, 0,
                    0, 0)
minFlowDryFut = (40612, 36000, 31835, 27223, 22760, 18149, 13686, 9074, 4463, 0, 0,
                    0)
minFlowCriticalFut = (13537, 12000, 10612, 9074, 7587, 6050, 4562, 3025, 1488, 0, 0, 0)
monthDemandFut = (24092, 23588, 23136, 22192, 21031, 19593, 15526, 10058, 4903, 0,
                    0, 0)
monthEvapFut = (4434, 4320, 4202, 400, 3679, 3176, 2454, 1571, 755, 0, 0, 0)

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
year = curTime.year()
mon = curTime.month()
day = curTime.day()
curStepMon = currentRuntimestep.month()
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime)

#Initialize the hydrologic index (hydroIndex)
hydroIndex = currentVariable.getPreviousValue(currentRuntimestep)

#bufferVal = buffer used by the water balance
#autoDryIndex = automatically set hydrologic index to dry if Lake Mendocino
#Storage drops below this value
#autoCriticalIndex = automatically set hydrologic index to critical if Lake
#Mendocino Storage drops below this value
bufferVal = 30000
autoDryIndex = 30000
autoCriticalIndex = 20000

#Get the current Lake Mendocino storage
LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
curLMStorage = LMstorageTS.getValue(step+1)

#Get the ensemble of Lake Mendocino inflows (from the 98-year period of record)
ensembleTS = currentVariable.localTimeSeriesGet("ensemble")

#Use the water balance approach from June 1 - Dec 1
j = mon
if mon < 10 and day == 1:
        z = 0
        countNormal = 0
        countDry = 0
        countCritical = 0
        lastIndex = 1
```

```
lastIndex2 = 1
#Use the water balance approach from June 1 - Dec 1 - the water balance is
#computed 98 times, using Lake Mendocino inflows from the ensemble
#dataset
while z < 99: #98 years in ensemble dataset
	i = (j - 1)
	counter = z * 12 + i
	endOfMonthStorage = 0
	endOfMonthDeficit = 0
	monthUsable = 0
	totalUsable = 0
	#Get the forecast
	forecastTS = currentVariable.localTimeSeriesGet("forecast")
	forecastedIndex = forecastTS.getValue(curTime)
	endofMonthStorageDummy = 0
	if forecastedIndex == 3: #Normal forecast
		while i < 9:
			#starting with the current month, loop through months
			#until September and compute the end of month
			#storage, storable water, and total usable water (for
			#the remainder of the water year)
			if j - 1 > i:
				endOfMonthStorage = 0
			elif j - 1 < i:
				endOfMonthStorage = endOfMonthStorage +
				ensembleTS.getValue(counter) - monthDemand[i] -
				monthEvap[i] - minFlowNormalInc[i]
			else:
				endOfMonthStorage = curLMStorage +
				ensembleTS.getValue(counter) - monthDemand[i] -
				monthEvap[i] - minFlowNormalInc[i]

			if endOfMonthStorage < 0:
				endOfMonthStorage = 0

			if endOfMonthStorage > endMonthStorageGC[i]:
				endOfMonthStorage = endMonthStorageGC[i]

			if endofMonthStorageDummy == 0:
				waterStore = endOfMonthStorage - curLMStorage
				endofMonthStorageDummy = endOfMonthStorage
			else:
				waterStore = endOfMonthStorage -
				endofMonthStorageDummy
				endofMonthStorageDummy = endOfMonthStorage

			if waterStore < 0:
				waterStore = 0

			monthUsable =  waterStore + monthDemand[i] +
			monthEvap[i] + minFlowNormalInc[i]
			if monthUsable > ensembleTS.getValue(counter):
				monthUsable = ensembleTS.getValue(counter)
```

```
                totalUsable += monthUsable

                i += 1
                counter += 1

elif forecastedIndex == 2: #Dry forecast
        while i < 9:
                #starting with the current month, loop through months
                #until September and compute the end of month
                #storage, storable water, and total usable water (for
                #the remainder of the water year)
                if j - 1 > i:
                        endOfMonthStorage = 0
                elif j - 1 < i:
                        endOfMonthStorage = endOfMonthStorage +
                        ensembleTS.getValue(counter) - monthDemand[i] -
                        monthEvap[i] - minFlowDryInc[i]
                else:
                        endOfMonthStorage = curLMStorage +
                        ensembleTS.getValue(counter) - monthDemand[i] -
                        monthEvap[i] - minFlowDryInc[i]

                if endOfMonthStorage < 0:
                        endOfMonthStorage = 0

                if endOfMonthStorage > endMonthStorageGC[i]:
                        endOfMonthStorage = endMonthStorageGC[i]

                if endofMonthStorageDummy == 0:
                        waterStore = endOfMonthStorage - curLMStorage
                        endofMonthStorageDummy = endOfMonthStorage
                else:
                        waterStore = endOfMonthStorage -
                        endofMonthStorageDummy
                        endofMonthStorageDummy = endOfMonthStorage

                if waterStore < 0:
                        waterStore = 0

                monthUsable =  waterStore + monthDemand[i] +
                monthEvap[i] + minFlowDryInc[i]
                if monthUsable > ensembleTS.getValue(counter):
                        monthUsable = ensembleTS.getValue(counter)

                totalUsable += monthUsable

                i += 1
                counter += 1

else: #Critical forecast
        while i < 9:
                #starting with the current month, loop through months
                #until September and compute the end of month
```

```
        #storage, storable water, and total usable water (for
        #the remainder of the water year)
        if j - 1 > i:
                endOfMonthStorage = 0
        elif j - 1 < i:
                endOfMonthStorage = endOfMonthStorage +
                ensembleTS.getValue(counter) - monthDemand[i] -
                monthEvap[i] - minFlowCriticalInc[i]
        else:
                endOfMonthStorage = curLMStorage +
                ensembleTS.getValue(counter) - monthDemand[i] -
                monthEvap[i] - minFlowCriticalInc[i]

        if endOfMonthStorage < 0:
                endOfMonthStorage = 0

        if endOfMonthStorage > endMonthStorageGC[i]:
                endOfMonthStorage = endMonthStorageGC[i]

        if endofMonthStorageDummy == 0:
                waterStore = endOfMonthStorage - curLMStorage
                endofMonthStorageDummy = endOfMonthStorage
        else:
                waterStore = endOfMonthStorage -
                endofMonthStorageDummy
                endofMonthStorageDummy = endOfMonthStorage

        if waterStore < 0:
                waterStore = 0

        monthUsable =  waterStore + monthDemand[i] +
        monthEvap[i] + minFlowCriticalInc[i]
        if monthUsable > ensembleTS.getValue(counter):
                monthUsable = ensembleTS.getValue(counter)

        totalUsable += monthUsable

        i += 1
        counter += 1

#Using the total usable water, compute the water available for
#meeting minimum flows
availableWater = totalUsable + curLMStorage - bufferVal
futureDemands = monthDemandFut[mon - 1] + monthEvapFut[mon - 1]
availableMinusDemands = availableWater - futureDemands
if (availableMinusDemands < 0):
        availableMinusDemands = 0

#Compare the available water to minimum flows to compute the
#hydrologic index – count the number of time each index is selected
#for each of the 98 ensemble datasets
if (availableMinusDemands > minFlowNormalFut[mon - 1]):
        countNormal += 1
elif (availableMinusDemands > minFlowDryFut[mon - 1]):
```

```
                countDry += 1
            else:
                countCritical += 1

            z += 1


        #Set the hydrologic index based on the count of normal, dry, and critical
        #from the ensemble datasets
        hydroIndex = 2
        if countCritical > 10:
            hydroIndex = 3
        elif countNormal > 60:
            hydroIndex = 1

#Use the auto-index option to set the hydrologic index (based on current storage)
if day == 1:
        if curLMStorage < autoCriticalIndex:
            hydroIndex = 3
        elif hydroIndex == 1 and curLMStorage < autoDryIndex:
            hydroIndex = 2

#Save the hydrologic index
currentVariable.setValue(currentRuntimestep, hydroIndex)
```

A.9 Lake Mendocino Storage Hydrologic Index – Evaluated Bi-Monthly, Five-Schedule Minimum Flows

The state variable code (script) for the Lake Mendocino storage, evaluated bi-monthly, hydrologic index is shown below. This script is similar to the previous storage hydrologic index alternatives; the difference is that the hydrologic index is evaluated bi-monthly and it includes the five-schedule minimum flow alternative. The script is evaluated by HEC-ResSim each time-step during the model simulation. Output from the script is the hydrologic index (referred to as hydroIndex). HEC-ResSim uses the hydrologic index when setting minimum flows from Lake Mendocino and Lake Sonoma. The "clean up" section of the state variable script has been omitted and only the computation section is included below.

```
#####
##### STATE VARIABLE SCRIPT INITIALIZATION SECTION
#####

from hec.script import Constants
from hec.hecmath import TimeSeriesMath, DSS, DSSFile
from hec.script import  ClientAppWrapper

#Load the storage thresholds – these are annual pattern time-series
def initStateVariable(currentVariable, network):

        tw=network.getRssRun().getCurrentComputeBlockRunTimeWindow()
        tws = tw.getTimeWindowString()
```

```
        global simulation
        module = ClientAppWrapper.getCurrentModule()
        simulation = module.getSimulation()
        simulationDssFilename = simulation.getOutputDSSFilePath()
        dssFile = "shared/ExternalTimeSeries.dss"
        parts = simulationDssFilename.split("/rss")
        directory, directory2 = parts
        dssFileName = '/'.join([directory, dssFile])
        DssFile = DSS.open(dssFileName, tws)

        storageCrit = DssFile.read("//LAKE MENDOCINO-STORAGECURVE/STOR-
        ZONE//1DAY/7/")
        storageCritTS = storageCrit.getData()
        currentVariable.localTimeSeriesNew("storageCrit", storageCritTS)

        storageDryCritical = DssFile.read("//LAKE MENDOCINO-STORAGECURVE/STOR-
        ZONE//1DAY/65/")
        storageDryCriticalTS = storageDryCritical.getData()
        currentVariable.localTimeSeriesNew("storageDryCritical", storageDryCriticalTS)

        storageDry = DssFile.read("//LAKE MENDOCINO-STORAGECURVE/STOR-
        ZONE//1DAY/6/")
        storageDryTS = storageDry.getData()
        currentVariable.localTimeSeriesNew("storageDry", storageDryTS)

        storageNormalDry = DssFile.read("//LAKE MENDOCINO-STORAGECURVE/STOR-
        ZONE//1DAY/55/")
        storageNormalDryTS = storageNormalDry.getData()
        currentVariable.localTimeSeriesNew("storageNormalDry", storageNormalDryTS)

        return Constants.TRUE

#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####

from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
curTime2 = curTime.clone()
curTime2.addDays(1) # this is necessary because simulation starts at 0000
day = curTime.day()
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime2)

#Initialize hydroIndex (Hydrologic Index)
hydroIndex = currentVariable.getPreviousValue(currentRuntimestep)

#Get storage thresholds – from initialization
#Compute hydrologic index by comparing current Lake Mendocino storage to
#thresholds (day → index is evaluated on the 1st and 16th of the month)
#notice this is the 5-schedule option
```

```
if day == 1 or day == 16:
        LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
        curLMStorage = LMstorageTS.getValue(step)
        storage5TS = currentVariable.localTimeSeriesGet("storageCrit")
        storage5 = storage5TS.getValue(curTime2)
        storage4TS = currentVariable.localTimeSeriesGet("storageDryCritical")
        storage4 = storage4TS.getValue(curTime2)
        storage3TS = currentVariable.localTimeSeriesGet("storageDry")
        storage3 = storage3TS.getValue(curTime2)
        storage2TS = currentVariable.localTimeSeriesGet("storageNormalDry")
        storage2 = storage2TS.getValue(curTime2)
        if curLMStorage < storage5:
                hydroIndex = 3
        elif curLMStorage < storage4:
                hydroIndex = 2.5
        elif curLMStorage < storage3:
                hydroIndex = 2
        elif curLMStorage < storage2:
                hydroIndex = 1.5
        else:
                hydroIndex = 1

#Save Hydrologic Index
currentVariable.setValue(currentRuntimestep, hydroIndex)
```

A.10 Future Available Water Hydrologic Index – Evaluated Bi-Monthly, Five-Schedule Minimum Flows

The state variable code (script) for the future available water hydrologic index is shown below. The script is evaluated by HEC-ResSim each time-step during the model simulation. Output from the script is the hydrologic index (referred to as hydroIndex). HEC-ResSim uses the hydrologic index when setting minimum flows from Lake Mendocino and Lake Sonoma. The "clean up" section of the state variable script has been omitted and only the computation section is included below.

```
#####
##### STATE VARIABLE SCRIPT INITIALIZATION SECTION
#####

from hec.script import Constants
from hec.hecmath import TimeSeriesMath, DSS, DSSFile
from hec.script import  ClientAppWrapper

#Load the forecasted inflow from the NWS ensemble, future available water
#thresholds, these are annual pattern time-series, and the hydrologic index from
#the Eel River watershed
def initStateVariable(currentVariable, network):

        tw=network.getRssRun().getRunTimeWindow()
        tws = tw.getTimeWindowString()

        global simulation
        module = ClientAppWrapper.getCurrentModule()
```

```
        simulation = module.getSimulation()
        simulationDssFilename = simulation.getOutputDSSFilePath()
        dssFile = "shared/ExternalTimeSeries.dss"
        parts = simulationDssFilename.split("/rss")
        directory, directory2 = parts
        dssFileName = '/'.join([directory, dssFile])
        DssFile = DSS.open(dssFileName, tws)

        forecast = DssFile.read("/UR/LAKE MENDOCINO/FLOWFORECASTACRE-
        FEET//1DAY/USGS4.2B/")
        forecastTS = forecast.getData()
        currentVariable.localTimeSeriesNew("forecast", forecastTS)

        forecastNum = DssFile.read("/RR/WATER SUPPLY/CONDITION//1DAY/RRSIM
        BASELINE/")
        forecastNumTS = forecastNum.getData()
        currentVariable.localTimeSeriesNew("forecastNum", forecastNumTS)

        AvailWatCrit = DssFile.read("//LAKE MENDOCINO-
        AVAILABLEWATER/WATER//1DAY/5/")
        AvailWatCritTS = AvailWatCrit.getData()
        currentVariable.localTimeSeriesNew("AvailWatCrit", AvailWatCritTS)

        AvailWatDryCrit = DssFile.read("//LAKE MENDOCINO-
        AVAILABLEWATER/WATER//1DAY/4/")
        AvailWatDryCritTS = AvailWatDryCrit.getData()
        currentVariable.localTimeSeriesNew("AvailWatDryCrit", AvailWatDryCritTS)

        AvailWatDry = DssFile.read("//LAKE MENDOCINO-
        AVAILABLEWATER/WATER//1DAY/3/")
        AvailWatDryTS = AvailWatDry.getData()
        currentVariable.localTimeSeriesNew("AvailWatDry", AvailWatDryTS)

        AvailWatNormalDry = DssFile.read("//LAKE MENDOCINO-
        AVAILABLEWATER/WATER//1DAY/2/")
        AvailWatNormalDryTS = AvailWatNormalDry.getData()
        currentVariable.localTimeSeriesNew("AvailWatNormalDry", AvailWatNormalDryTS)

        AvailWatNormal = DssFile.read("//LAKE MENDOCINO-
        AVAILABLEWATER/WATER//1DAY/1/")
        AvailWatNormalTS = AvailWatNormal.getData()
        currentVariable.localTimeSeriesNew("AvailWatNormal", AvailWatNormalTS)

        return Constants.TRUE

#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####

from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS
```

```
#The following variables are used by the empirical relationship of future available
#water to hydrologic conditions, inflows from the Potter Valley Project, Lake
#Mendocino conservation storage, and constants developed by SCWA
monthList = ("JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC")
VarC = (1.08, 1.16, 1.20, 1.29, 1.44, 1.52, 1.59, 1.83, 1.92, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.46,
        0.53, 0.65, 0.73, 0.86, 0.97)
VarB = (14410, 11140, 11980, 9961, 8721, 4476, 1970, 180, 180, 180, 54, 13, 0, 0, 0, 0,
        0, 0, 57400, 53600, 46960, 37780, 26510, 19830)
LMMS = (68409, 68409, 68409, 68409, 68409, 77214, 86828, 95997, 105305, 110967,
        110967, 110967, 110967, 110967, 110967, 110967, 110967, 110967, 108117,
        87200, 68409, 68409, 68409, 68409)
PCD1Normal = (23552, 19532, 15202, 11148, 7094, 4949, 3203, 2162, 1121, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 45840, 43248, 39069, 35002, 31175, 27460)
PCD1Dry = (17730, 15838, 13048, 9744, 6045, 4315, 2599, 1576, 802, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 37326, 34678, 30553, 27001, 23884, 20981)
PCD1Critical = (5800, 4961, 2885, 2222, 964, 815, 447, 298, 149, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        10645, 10496, 10337, 10188, 10039, 9051)
PCD2Normal = (21207, 21207, 21207, 21207, 21207, 21207, 21207, 21207, 21207,
        21207, 18827, 16644, 14461, 12278, 9871, 7499, 4859, 2293, 21207, 21207,
        21207, 21207, 21207, 21207)
PCD2Dry = (10573, 10573, 10573, 10573, 10573, 10573, 10573, 10573, 10573, 10573,
        9535, 8791, 8047, 7303, 6252, 5092, 3641, 2280, 10573, 10573, 10573, 10573,
        10573, 10573)
PCD2Critical = (1371, 1371, 1371, 1371, 1371, 1371, 1371, 1371, 1371, 1371, 1212,
        1063, 914, 765, 606, 457, 298, 149, 1371, 1371, 1371, 1371, 1371, 1371)

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
curTime2 = curTime.clone()
curTime2.addDays(1) # this is necessary because simulation starts at 0000
year = curTime.year()
mon = curTime.month()
day = curTime.day()
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime2)
#period2Forecast = HecTime()
#period2Forecast.setYearMonthDay(year, 5, 16, 0) # 0 is minutes past midnight

#Initialize the hydrologic index
hydroIndex = currentVariable.getPreviousValue(currentRuntimestep)

#Compute the future available water on the 1st of the month
j = mon
if day == 1:
        LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
        curLMStorage = LMstorageTS.getValue(step)
        futureAvailWater = 0
        forecastTS = currentVariable.localTimeSeriesGet("forecast")
        #forecastedInflow2 = forecastTS.getValue(period2Forecast)
        forecastedInflow1 = forecastTS.getValue(curTime2)
        forecastLakePillsNumTS = currentVariable.localTimeSeriesGet("forecastNum")
        forecastedLakePillsNumIndex = forecastLakePillsNumTS.getValue(curTime2)
        i = (j-1)*2
        if forecastedLakePillsNumIndex == 1:
```

```
        PCD1 = PCD1Normal[i]
        PCD2 = PCD2Normal[i]
    elif forecastedLakePillsNumIndex == 2:
        PCD1 = PCD1Dry[i]
        PCD2 = PCD2Dry[i]
    else:
        PCD1 = PCD1Critical[i]
        PCD2 = PCD2Critical[i]

    if curLMStorage > LMMS[i]:
        curLMStorage = LMMS[i]
    VarA = VarC[i] - (curLMStorage / LMMS[i])
    if VarA > 1:
        VarA = 1
    if mon > 5 and mon < 10:
        VarA = 0
    futureAvailableWater = VarA * (forecastedInflow1 + PCD1) + PCD2 + curLMStorage
    + VarB[i]

    availWat5TS = currentVariable.localTimeSeriesGet("AvailWatCrit")
    HI5 = availWat5TS.getValue(curTime2)
    availWat4TS = currentVariable.localTimeSeriesGet("AvailWatDryCrit")
    HI4 = availWat4TS.getValue(curTime2)
    availWat3TS = currentVariable.localTimeSeriesGet("AvailWatDry")
    HI3 = availWat3TS.getValue(curTime2)
    availWat2TS = currentVariable.localTimeSeriesGet("AvailWatNormalDry")
    HI2 = availWat2TS.getValue(curTime2)
    availWat1TS = currentVariable.localTimeSeriesGet("AvailWatNormal")
    HI1 = availWat1TS.getValue(curTime2)
    if futureAvailableWater < HI5:
        hydroIndex = 3
    elif futureAvailableWater < HI4:
        hydroIndex = 2.5
    elif futureAvailableWater < HI3:
        hydroIndex = 2
    elif futureAvailableWater < HI2:
        hydroIndex = 1.5
    else:
        hydroIndex = 1

#Compute the future available water on the 1st of the month
elif day == 16:
    LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool", "Stor")
    curLMStorage = LMstorageTS.getValue(step)
    futureAvailWater = 0
    forecastTS = currentVariable.localTimeSeriesGet("forecast")
    forecastedInflow1 = forecastTS.getValue(curTime2)
    #forecastedInflow2 = forecastTS.getValue(period2Forecast)
    forecastLakePillsNumTS = currentVariable.localTimeSeriesGet("forecastNum")
    forecastedLakePillsNumIndex = forecastLakePillsNumTS.getValue(curTime2)
    i = (j-1)*2+1
    if forecastedLakePillsNumIndex == 1:
        PCD1 = PCD1Normal[i]
        PCD2 = PCD2Normal[i]
```

```
	elif forecastedLakePillsNumIndex == 2:
		PCD1 = PCD1Dry[i]
		PCD2 = PCD2Dry[i]
	else:
		PCD1 = PCD1Critical[i]
		PCD2 = PCD2Critical[i]

	if curLMStorage > LMMS[i]:
		curLMStorage = LMMS[i]
	VarA = VarC[i] - (curLMStorage / LMMS[i])
	if VarA > 1:
		VarA = 1
	if mon > 5 and mon < 10:
		VarA = 0
	futureAvailableWater = VarA * (forecastedInflow1 + PCD1) + PCD2 + curLMStorage
	+ VarB[i]

	availWat5TS = currentVariable.localTimeSeriesGet("AvailWatCrit")
	HI5 = availWat5TS.getValue(curTime2)
	availWat4TS = currentVariable.localTimeSeriesGet("AvailWatDryCrit")
	HI4 = availWat4TS.getValue(curTime2)
	availWat3TS = currentVariable.localTimeSeriesGet("AvailWatDry")
	HI3 = availWat3TS.getValue(curTime2)
	availWat2TS = currentVariable.localTimeSeriesGet("AvailWatNormalDry")
	HI2 = availWat2TS.getValue(curTime2)
	availWat1TS = currentVariable.localTimeSeriesGet("AvailWatNormal")
	HI1 = availWat1TS.getValue(curTime2)
	if futureAvailableWater < HI5:
		hydroIndex = 3
	elif futureAvailableWater < HI4:
		hydroIndex = 2.5
	elif futureAvailableWater < HI3:
		hydroIndex = 2
	elif futureAvailableWater < HI2:
		hydroIndex = 1.5
	else:
		hydroIndex = 1

#Save Hydrologic Index
currentVariable.setValue(currentRuntimestep, hydroIndex)
```

A.11 Modified Storage Hydrologic Index – Evaluated Bi-Monthly, Five-Schedule Minimum Flows

The state variable code (script) for the modified storage hydrologic index is shown below. The script is evaluated by HEC-ResSim each time-step during the model simulation. Output from the script is the hydrologic index (referred to as hydroIndex). HEC-ResSim uses the hydrologic index when setting minimum flows from Lake Mendocino and Lake Sonoma. The "clean up" section of the state variable script has been omitted and only the computation section is included below.

```
#####
##### STATE VARIABLE SCRIPT INITIALIZATION SECTION
#####
```

```
from hec.script import Constants
from hec.hecmath import TimeSeriesMath, DSS, DSSFile
from hec.script import  ClientAppWrapper

def initStateVariable(currentVariable, network):

        tw=network.getRssRun().getRunTimeWindow()
        tws = tw.getTimeWindowString()

        global simulation
        module = ClientAppWrapper.getCurrentModule()
        simulation = module.getSimulation()
        simulationDssFilename = simulation.getOutputDSSFilePath()
        dssFile = "shared/ExternalTimeSeries.dss"
        parts = simulationDssFilename.split("/rss")
        directory, directory2 = parts
        dssFileName = '/'.join([directory, dssFile])
        DssFile = DSS.open(dssFileName, tws)

#PVP Current Water Supply Condition
        forecastNum = DssFile.read("/RR/PVP OPS/CONDITION//1DAY/RRSIM BASELINE/")
        forecastNumTS = forecastNum.getData()
        currentVariable.localTimeSeriesNew("forecastNum", forecastNumTS)

#Storage plus West Fork & PCD Flows Input Variables
        LMStorQwfSch2 = DssFile.read("//LMSTOR QWF PCD BIMONTH/WATER//1DAY/2/")
        LMStorQwfSch2TS = LMStorQwfSch2.getData()
        currentVariable.localTimeSeriesNew("LMStorQwfSch2", LMStorQwfSch2TS)

        LMStorQwfSch3 = DssFile.read("//LMSTOR QWF PCD BIMONTH/WATER//1DAY/3/")
        LMStorQwfSch3TS = LMStorQwfSch3.getData()
        currentVariable.localTimeSeriesNew("LMStorQwfSch3", LMStorQwfSch3TS)

        LMStorQwfSch4 = DssFile.read("//LMSTOR QWF PCD BIMONTH/WATER//1DAY/4/")
        LMStorQwfSch4TS = LMStorQwfSch4.getData()
        currentVariable.localTimeSeriesNew("LMStorQwfSch4", LMStorQwfSch4TS)

        LMStorQwfSch5 = DssFile.read("//LMSTOR QWF PCD BIMONTH/WATER//1DAY/5/")
        LMStorQwfSch5TS = LMStorQwfSch5.getData()
        currentVariable.localTimeSeriesNew("LMStorQwfSch5", LMStorQwfSch5TS)

#Get two week cumulative flow at the E. Fork Russian River
        CumQwf = DssFile.read("//TOT QWF BIMONTH/WATER//1DAY/USGS4.2B/")
        CumQwfTS = CumQwf.getData()
        currentVariable.localTimeSeriesNew("CumQwf", CumQwfTS)

        return Constants.TRUE

#####
##### STATE VARIABLE SCRIPT COMPUTATION SECTION
#####
```

```
from hec.heclib.util import HecTime
from hec.script import ClientAppWrapper
from hec.hecmath import DSS

#Input data for modified storage, inflow from the Potter Valley Project for Normal,
Dry, or #Critical Conditions in the Eel River watershed
monthList = ("JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC")
LMMS = (68409, 68409, 68409, 68409, 68409, 77214, 86828, 95997, 105305, 110967,
        110967, 110967, 110967, 110967, 110967, 110967, 110967, 110967, 108117,
        87200, 68409, 68409, 68409, 68409)
PCD1Normal = (0, 0, 0, 0, 0, 0, 22542, 21501, 20460, 19339, 16959, 14727, 12496,
        10265, 7884, 5653, 3273, 1041, 0, 0, 0, 0, 0,0)
PCD1Dry = (0, 0, 0, 0, 0, 0, 9650, 8628, 7884, 7141, 6347, 5603, 4860, 4116, 3322,
        2579, 1785, 1041, 0, 0, 0, 0, 0, 0)
PCD1Critical = (0, 0, 0, 0, 0, 0, 1815, 1666, 1517, 1369, 1210, 1061, 912, 764, 605, 456,
        298, 149, 0, 0, 0, 0, 0, 0)

#Set time variables for current time step
curTime = currentRuntimestep.getHecTime()
curTime2 = curTime.clone()
curTime2.addDays(1) # this is necessary because simulation starts at 0000
year = curTime.year()
mon = curTime.month()
day = curTime.day()
step = currentRuntimestep.getRunTimeWindow().getStepAtTime(curTime2)

#Get hydro index from previous step
hydroIndex = currentVariable.getPreviousValue(currentRuntimestep)

#Compute the hydrologic index using current Lake Mendocino Storage and the two
#week cumulative flow on the W. Fork Russian River
j = mon
if mon < 4 or mon > 9:
        if day == 1:
                LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool",
                "Stor")
                curLMStorage = LMstorageTS.getValue(step)
                perCumQwfTS = currentVariable.localTimeSeriesGet("CumQwf")
                curCumQwf = perCumQwfTS.getValue(curTime2)
                i = (j-1)*2
                if curLMStorage > LMMS[i]:
                        curLMStorage = LMMS[i]

                curLMStorQwf = curLMStorage + curCumQwf

                sch5LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch5")
                HI5 = sch5LMStorQwfTS.getValue(curTime2)
                sch4LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch4")
                HI4 = sch4LMStorQwfTS.getValue(curTime2)
                sch3LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch3")
                HI3 = sch3LMStorQwfTS.getValue(curTime2)
                sch2LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch2")
                HI2 = sch2LMStorQwfTS.getValue(curTime2)
```

```
            if curLMStorQwf < HI5:
                hydroIndex = 3
            elif curLMStorQwf < HI4:
                hydroIndex = 2.5
            elif curLMStorQwf < HI3:
                hydroIndex = 2
            elif curLMStorQwf < HI2:
                hydroIndex = 1.5
            else:
                hydroIndex = 1

        elif day == 16:
            LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool",
            "Stor")
            curLMStorage = LMstorageTS.getValue(step)
            perCumQwfTS = currentVariable.localTimeSeriesGet("CumQwf")
            curCumQwf = perCumQwfTS.getValue(curTime2)
            i = (j-1)*2+1
            if curLMStorage > LMMS[i]:
                curLMStorage = LMMS[i]

            curLMStorQwf = curLMStorage + curCumQwf

            sch5LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch5")
            HI5 = sch5LMStorQwfTS.getValue(curTime2)
            sch4LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch4")
            HI4 = sch4LMStorQwfTS.getValue(curTime2)
            sch3LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch3")
            HI3 = sch3LMStorQwfTS.getValue(curTime2)
            sch2LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch2")
            HI2 = sch2LMStorQwfTS.getValue(curTime2)

            if curLMStorQwf < HI5:
                hydroIndex = 3
            elif curLMStorQwf < HI4:
                hydroIndex = 2.5
            elif curLMStorQwf < HI3:
                hydroIndex = 2
            elif curLMStorQwf < HI2:
                hydroIndex = 1.5
            else:
                hydroIndex = 1

#Compute the hydrologic index using current Lake Mendocino Storage, the two
#week cumulative flow on the W. Fork Russian River, and Potter Valley Project
#water
elif mon>3 and mon<6:
        if day == 1:
            LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool",
            "Stor")
            curLMStorage = LMstorageTS.getValue(step)
            perCumQwfTS = currentVariable.localTimeSeriesGet("CumQwf")
            curCumQwf = perCumQwfTS.getValue(curTime2)
            forecastLakePillsNumTS = currentVariable.localTimeSeriesGet("forecastNum")
```

```
		forecastedLakePillsNumIndex = forecastLakePillsNumTS.getValue(curTime2)
		i = (j-1)*2
		if forecastedLakePillsNumIndex == 1:
			PCD1 = PCD1Normal[i]
		elif forecastedLakePillsNumIndex == 2:
			PCD1 = PCD1Dry[i]
		else:
			PCD1 = PCD1Critical[i]

		if curLMStorage > LMMS[i]:
			curLMStorage = LMMS[i]

		curLMStorQwfPCD = curLMStorage + PCD1 + curCumQwf

		sch5LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch5")
		HI5 = sch5LMStorQwfTS.getValue(curTime2)
		sch4LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch4")
		HI4 = sch4LMStorQwfTS.getValue(curTime2)
		sch3LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch3")
		HI3 = sch3LMStorQwfTS.getValue(curTime2)
		sch2LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch2")
		HI2 = sch2LMStorQwfTS.getValue(curTime2)

		if curLMStorQwfPCD < HI5:
			hydroIndex = 3
		elif curLMStorQwfPCD < HI4:
			hydroIndex = 2.5
		elif curLMStorQwfPCD < HI3:
			hydroIndex = 2
		elif curLMStorQwfPCD < HI2:
			hydroIndex = 1.5
		else:
			hydroIndex = 1

	elif day == 16:
		LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool",
		"Stor")
		curLMStorage = LMstorageTS.getValue(step)
		perCumQwfTS = currentVariable.localTimeSeriesGet("CumQwf")
		curCumQwf = perCumQwfTS.getValue(curTime2)
		forecastLakePillsNumTS = currentVariable.localTimeSeriesGet("forecastNum")
		forecastedLakePillsNumIndex = forecastLakePillsNumTS.getValue(curTime2)
		i = (j-1)*2+1
		if forecastedLakePillsNumIndex == 1:
			PCD1 = PCD1Normal[i]
		elif forecastedLakePillsNumIndex == 2:
			PCD1 = PCD1Dry[i]
		else:
			PCD1 = PCD1Critical[i]

		if curLMStorage > LMMS[i]:
			curLMStorage = LMMS[i]

		curLMStorQwfPCD = curLMStorage + PCD1 + curCumQwf
```

```
            sch5LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch5")
            HI5 = sch5LMStorQwfTS.getValue(curTime2)
            sch4LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch4")
            HI4 = sch4LMStorQwfTS.getValue(curTime2)
            sch3LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch3")
            HI3 = sch3LMStorQwfTS.getValue(curTime2)
            sch2LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch2")
            HI2 = sch2LMStorQwfTS.getValue(curTime2)

            if curLMStorQwfPCD < HI5:
                  hydroIndex = 3
            elif curLMStorQwfPCD < HI4:
                  hydroIndex = 2.5
            elif curLMStorQwfPCD < HI3:
                  hydroIndex = 2
            elif curLMStorQwfPCD < HI2:
                  hydroIndex = 1.5
            else:
                  hydroIndex = 1

elif mon==6:
      if day == 1:
            LMstorageTS = network.getTimeSeries("Reservoir","Lake Mendocino", "Pool",
            "Stor")
            curLMStorage = LMstorageTS.getValue(step)
            perCumQwfTS = currentVariable.localTimeSeriesGet("CumQwf")
            curCumQwf = perCumQwfTS.getValue(curTime2)
            forecastLakePillsNumTS = currentVariable.localTimeSeriesGet("forecastNum")
            forecastedLakePillsNumIndex = forecastLakePillsNumTS.getValue(curTime2)
            i = (j-1)*2
            if forecastedLakePillsNumIndex == 1:
                  PCD1 = PCD1Normal[i]
            elif forecastedLakePillsNumIndex == 2:
                  PCD1 = PCD1Dry[i]
            else:
                  PCD1 = PCD1Critical[i]

            if curLMStorage > LMMS[i]:
                  curLMStorage = LMMS[i]

            curLMStorQwfPCD = curLMStorage + PCD1 + curCumQwf

            sch5LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch5")
            HI5 = sch5LMStorQwfTS.getValue(curTime2)
            sch4LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch4")
            HI4 = sch4LMStorQwfTS.getValue(curTime2)
            sch3LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch3")
            HI3 = sch3LMStorQwfTS.getValue(curTime2)
            sch2LMStorQwfTS = currentVariable.localTimeSeriesGet("LMStorQwfSch2")
            HI2 = sch2LMStorQwfTS.getValue(curTime2)

            if curLMStorQwfPCD < HI5:
                  hydroIndex = 3
```

```
        elif curLMStorQwfPCD < HI4:
                hydroIndex = 2.5
        elif curLMStorQwfPCD < HI3:
                hydroIndex = 2
        elif curLMStorQwfPCD < HI2:
                hydroIndex = 1.5
        else:
                hydroIndex = 1

#Save the Hydrologic Index
currentVariable.setValue(currentRuntimestep, hydroIndex)
```