APPENDIX B –

Finalized Inflow Data Development

Section 1. Introduction

This report provides a detailed account of the inflow development for the Tar River Basin Hydrologic Model. The inflow record runs from January 1930 to September 2009[1]. This period is designed to capture as many drought events as possible, including the extreme droughts in the early 1930s and 2000s. There are 11 streamflow gages in the basin that are used in this project. These are listed in Table 1. These gages have at least 10 years of daily data with which to make valid statistical comparisons with other gages. Most of the gages have incomplete records; they either started after 1930 or ended before 2009. Some of the gages were used just to provide more data for *fillin* (see below) when computing statistics.

The inflow dataset is based on "unimpaired" gage flows. Gages only show the actual flow in the stream; they have no information about what the flow would have been without human intervention. "Impairments" are modifications of the natural flows due to change in reservoir storage (including evaporation and precipitation on the reservoir surface) and consumptive withdrawals of water (municipal, industrial, or agricultural). If water is withdrawn above a gage and returned to the river below the gage, the impairment is the entire withdrawal.

The next section describes the process used to compute daily flows and gains. Because of the noise in the data, it is important to look at the data at each step to find unrealistic values. These are noted later.

[1] A provisional record extends beyond this date, but this does not account for most of the actual impairments. Future updates will require impairment data for the inflow dataset to be considered finalized.

Section 2. Data and General Procedure

The first step in building the record is to compute the unimpaired gage flows. These computations are contained in the spreadsheet in the ***inflow_unimpairment.xls*** of the gage data directory. The unimpaired gage data is summarized in the ***unimpaired_summary.xls*** file. Impairments in the basin accumulate as each downstream gage is included. For example, the impairments upstream of Tarboro include the impairments on the Tar River (including the operations of Tar River Reservoir) and Fishing Creek. The impairment is calculated as follows:

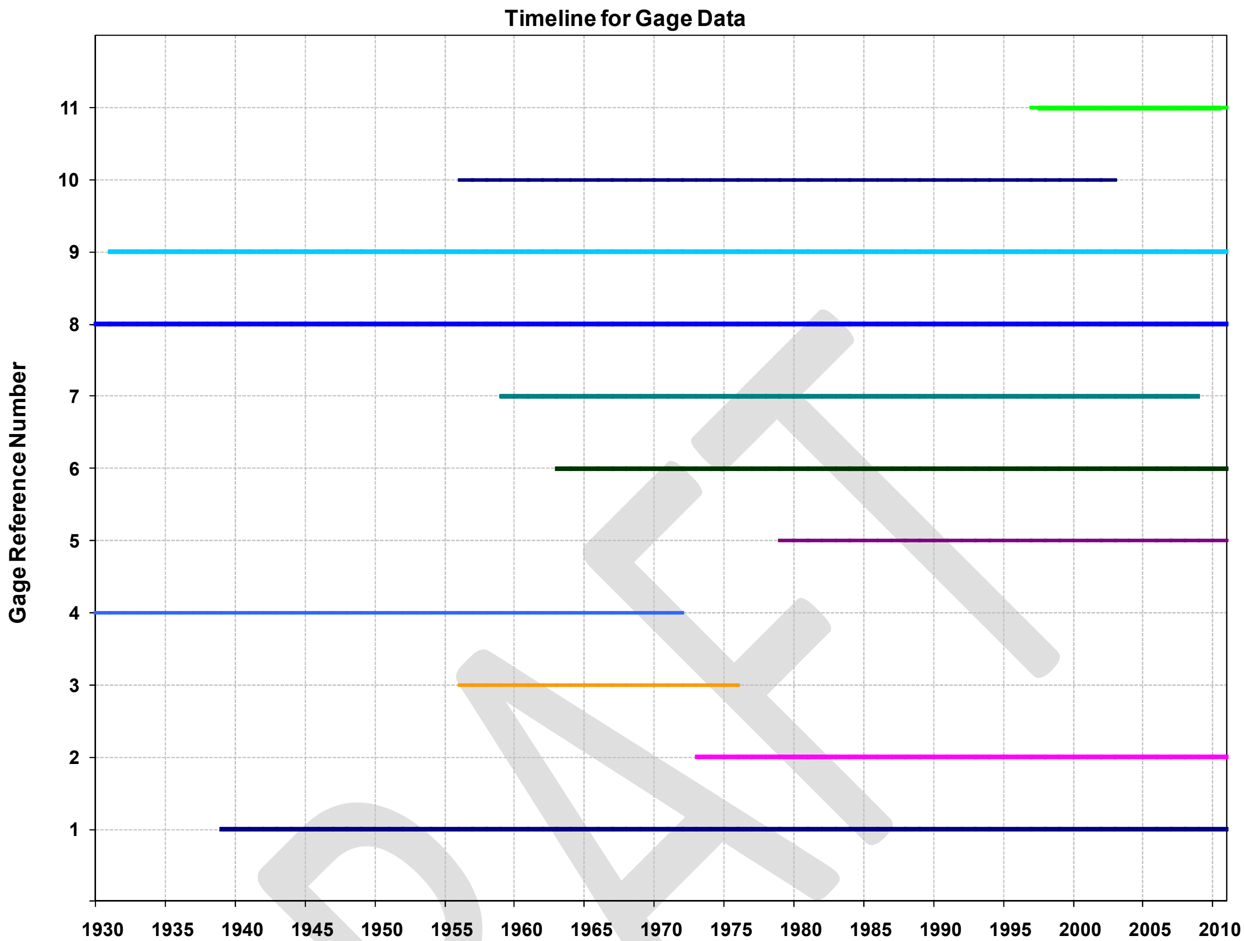
Unimpaired gage flow = gage flow + upstream water withdrawal (by agricultural, municipal, and industrial users) – upstream discharge (water or wastewater from municipal or industrial users) + upstream change in reservoir storage + upstream evaporation on the reservoir surface - upstream precipitation on the reservoir surface.

The discharge and withdrawal data were obtained by Moffatt and Nichol. The data are based on monthly average demand for recent years, and extrapolated back to 1930 using population census data. The reservoir operations data for Tar River Reservoir were obtained from the Rocky Mount water utility.

Evaporation and precipitation data were also collected by HydroLogics. Evaporation data are based on daily measurements from the Chapel Hill station, which is the only long-term station in the region. When measurements are missing, the long-term monthly averages are used. Precipitation data from the station closest to a reservoir are used to estimate the precipitation at the reservoir. When these data are missing, data from the next nearest station are used, and so on. For each reservoir, HydroLogics calculated a daily timeseries of net evaporation (or the difference between evaporation and precipitation) for the hydrologic record. These data are contained in spreadsheets in the “Evap-Precip” folder. These data are used to (1) estimate the historic change in reservoir storage due to net evaporation and (2) estimate net evaporation on the reservoir surface during OASIS model simulation.

Table 1. List of Gages

USGS Number	Description	Period of Record	Ref No.	Ref. Name	Drain. Area
02081500	TAR RIVER NEAR TAR RIVER, NC	10/1939-present	1	Tarr	167
02081747	TAR R AT US 401 AT LOUISBURG, NC	10/1973-present	2	Loui	427
02081800	CEDAR CREEK NEAR LOUISBURG, NC	10/1956-09/1975	3	Cedr	47.8
02082000	TAR RIVER NEAR NASHVILLE, NC	10/1928-01/1971	4	Nash	701
02082506	TAR R BL TAR R RESERVOIR NR ROCKY MOUNT, NC	08/1972-present	5	Resv	777
02082770	SWIFT CREEK AT HILLIARDSTON, NC	08/1963-present	6	Swif	166
02082950	LITTLE FISHING CREEK NEAR WHITE OAK, NC	10/1959-present	7	Lfsh	177
02083000	FISHING CREEK NEAR ENFIELD, NC	10/1926-present	8	Fish	526
02083500	TAR RIVER AT TARBORO, NC	10/1931-present	9	Tarb	2,183
02083800	CONETOE CREEK NEAR BETHEL, NC	12/1956-06/2002	10	Cone	78.1
02084000	TAR RIVER AT GREENVILLE, NC	04/1997-present	11	Gree	2,660



The second step in the inflow development process is to fill in the missing flows and gains for each gage with missing records. This requires assembling a monthly record of unimpaired flows and gains based on the daily unimpaired data computed above. These flows and gains are fed into a program named *fillin* (developed by William Alley and Alan Burns of the USGS[2]). We will refer to these as "extended" flows and gains. This is done on a monthly basis because *fillin* only works with monthly data. The gages associated with the flows and gains used in the remainder of this document are shown in Table 2. The third step is to apportion the extended flows and gains to make sure that their volumes match downstream unimpaired gage flows. The monthly flows and gains are then disaggregated into daily values using local, unimpaired gages. These steps are described in detail in Section 3.

The last step in the process is to compute the OASIS nodal inflows based on the flows and gains computed above. This step is described in detail in Section 4.

For the Tar River near Tar River gage (Tarr), *fillin* was used to complete its missing record from

[2] "Mixed-Station Extension of Monthly Streamflow Records," *Journal of Hydraulic Engineering*, ASCE, Vol. 109, No. 10, October 1983.

January 1930 to September 1939.

For the Tar River at Louisburg gage (Loui), *fillin* was used to complete its missing record from January 1930 to September 1975.

For the Cedar Creek near Louisburg gage (Cedr), *fillin* was used to complete its missing record from January 1930 to September 1956, and from October 1975 to September 2009.

Because the Tar River Reservoir is a key location in the basin, we assigned inflows to this node (with a drainage area of 775 square miles). The inflows are based on a drainage area adjustment of the upstream Nashville gage (701 square miles) from 1930 to 1971, and are back-calculated using reservoir operating data 1979 to 2009. Gaps from 1971 to 1978 in the reservoir inflow record are completed using *fillin*. The reservoir location (synthetic gage) is given the name Resv.

For the Swift Creek near Hilliardston gage (Swif), *fillin* was used to complete its missing record from January 1930 to July 1963.

For the Fishing Creek near White Oak gage (Lfsh), *fillin* was used to complete its missing record from January 1930 to September 1959.

The records of the gage Tar River at Tarboro start in March 1932. We extended the record using *fillin* back to January 1930.

For the Conetoe Creek near Bethel gage (Cone), *fillin* was used to complete its missing record from January 1930 to November 1956 and from July 2002 to September 2009.

For the Tar River near Greenville gage (Gree), *fillin* was used to complete its missing record from January 1930 to March 1997.

Table 2. Gages Where Flows and Gains Are Computed

Gage	**Flow or Gain**	**Gain Calculation**
Tarr	Flow	
Loui	Gain	Loui flow – Tarr flow
Resv	Gain	Resv Flow - Loui flow – Cedr flow
Swif	Flow	
Lfsh	Flow	
Fish	Gain	Fish flow – Lfsh flow
Tarb	Gain	Tarb Flow – Resv Flow – Swif Flow – Fish Flow
Cone	Flow	
Gree	Gain	Gree flow – Tarb flow – Cone Flow

Section 3. Computation of Extended Gage Flows and Reach Gains

All the computations outlined in this section are done on monthly data, which reduces noise and is required for statistical hydrology programs like *fillin*. Noisy data occurs when time of travel differences occur or when the impairment data create artificial variation in the flows.

First, the actual gains are determined from the unimpaired gage flows. These computations are done with the DSSVue script, ***compute_gain.py.*** Next, *fillin* is run to compute the extended flows and gains for the gages with missing records. Note that *fillin* preserves the actual flows and gains where they exist. These extended flows and gains are then "scaled." The objective of scaling is to ensure that the sum of filled-in flows upstream of a gage with an actual record equals the actual unimpaired flow at that gage. The *fillin* program does not ensure this for two reasons. First, it utilizes only a single correlated record for each value generated, thus ignoring sums, and second, it works with log transforms, and not actual flows.

Here is an example. We want to compute the unimpaired gain at the Louisburg gage (Loui) when the gage data is missing. We extended the gain at Loui using *fillin*. Now we want to adjust those extended values so that the sum of the flows and gains down to the reservoir (Resv) match the unimpaired flow at the reservoir. So we say that we maintain the Resv flow less the inflows to Cedr and Tarr inflows by scaling with the sum of the Loui and Resv extended gains. The calculation is:

> Scaled Loui extended gain = (Resv flow – Cedr flow – Tarr flow)* (Loui extended gain) / (Loui extended gain + Resvs extended gain)

The companion calculation for the flow at Resv is:

> Scaled Resv extended gain = (Resv flow – Cedr flow – Tarr flow)* (Resv extended gain) / (Loui extended gain + Resvs extended gain)

Thus the sum of the actual unimpaired flows (Tarr and Cedr) and the scaled gains (Loui and Resv) equals the unimpaired gage flow at Resv.

In this way we ensure that the total volume of all the flows and gains, be they actual or extended, upstream from a given gage match the unimpaired flow at the gage, preserving the unimpaired gaged flows.

The DSSVue script used to do these calculations is ***scale_flow_gain.py***. The output is the file ***scale_flow_gain.dss***, which contains monthly flows and gains.

The next step is to disaggregate the monthly flows into daily flows. This is done using flows for a daily, unimpaired gage that is local or has similar drainage area (call it a "reference gage"). We multiply the monthly value by the ratio of that day's flow to that month's flow at the reference gage. The disaggregation formula is:

daily ratio = daily reference value / monthly reference value
daily computed value = monthly computed value * daily ratio

Two DSSVue scripts are used to do this step: ***convert_month_to_day.py*** and ***disaggregate.py***. DSSVue cannot work on two records with different time steps, so the first step is to convert the monthly value to daily. For example, the gain at Resv for January 1930 is 156 cfs; converting the monthly flow to daily gives 156 cfs for each day in the month. The first script converts the monthly flows to daily flows for each location where we need them. The second script computes the daily flows and gains as shown above.

It is important to note that we are not trying to replicate history in computing the OASIS inflows; rather, we are trying to build daily flows whose variation is *representative* of history while preserving unimpaired gaged flows as "ground truth".

Note that actual daily values of unimpaired gage *flows* are often maintained in the script files. Actual daily *gains* between gages are generally not maintained due to their "noise", so the script files aggregate them monthly and then disaggregate them back to daily values using a locally unimpaired gage. Therefore, actual flows and gains on a monthly basis are maintained, but generally only the former will be maintained on a daily basis.

The remainder of this section shows the details in how the records were extended and disaggregated. This is the short hand used in the remainder of this document:

d/s = downstream
u/s = upstream
DA = drainage area
F = actual or scaled flow at a gage
XF = "extended" flow at a gage as computed by *fillin* when actual flows do not exist
G = actual or scaled gain, or inflow, between two locations, which is the difference of u/s gage F and d/s gage F
XG = extended gain between two locations as computed by *fillin* when actual gains do not exist

Tar River River near Tar River FLOW (Tarr), 10/39– present

10/28 – 09/39 TarF = (ResvF) * TarXF / (TarXF + LouiXG + CedrXF + ResvXG)
10/39 – 09/09 use actual daily values
Disaggregate filled in values with Fish

Tar River at Louisburg GAIN (Loui), 10/73 – present

10/28 – 09/39 LouiG = (ResvF) * LouiXG / (TarXF + LouiXG + CedrXF + ResvXG)
10/39 – 09/56 LouiG = (ResvF - TarF) * LouiXG / (LouiXG + CedrXF + ResvXG)
10/56 – 01/71 LouiG = (ResvF – CedrF – TarF) * LouiXG / (LouiXG + ResvXG)
02/71 – 09/73 LouiG = (TarbF – FishF – SwifF – CedrF – TarF) * LouiXF / (LouiXG + ResvXG + TarbXG)
10/73 – 09/09 use actual monthly values, disaggregate with Swif
Disaggregate whole record with Fish or Swif when available

Cedar Creek near Louisburg FLOW (10/56 – 09/1975)

10/28 – 09/39   CedrF = (ResvF) * CedrXF / (TarXF + LouiXG + CedrXF + ResvXG)
10/39 – 09/56 CedrF = (ResvF - TarF) * CedrXF/ (LouiXG + CedrXF + ResvXG)
10/56 – 09/75 use actual daily values
10/75 – 12/78 CedrF = (TarbF – FishF – SwifF – LouiF – TarF) * CedrF / (CedrXF + ResvXG + TarbXG)
1/79 - present CedrF = (ResvF – LouiF – TarF) * CedrXF/ (CedrXF + ResvXG)
Disaggregate filled in values with Fish or Swif when available

Tar River Reservoir GAIN (Resv), 10/28 – 01/71, 01/79 - present
10/28 – 09/39 ResvG = (ResvF - TarrF) * ResvXG / (LouiXG + CedrXF + ResvXG)
10/39 – 09/56 ResvG = (ResvF - TarrF - CedrF) * ResvXG / (LouiXG + ResvXG)
10/56 – 01/71 ResvG = (ResvF - TarrF - CedrF) * ResvXG / (LouiXG + ResvXG)
02/71 – 09/73 ResvG = (TarbF – FishF – SwifF – CedrF – TarF) * ResvXG / (LouiXG + ResvXG + TarbXG)
10/73 – 09/75   ResvG = (TarbF – FishF – SwifF – CedrF – LouiG – TarF) * ResvXG / (ResvXG + TarbXG)
10/75 – 12/78   ResvG = (TarbF – FishF – SwifF – LouiF – TarF) * ResvG / (CedrXF + ResvXG + TarbXG)
1/79 - present ResvG = (ResvF – LouiF – TarF) * ResvXG/ (CedrXF + ResvXG)
Disaggregate whole record with Fish

Swift Creek at Hilliardston FLOW (Swif), 08/63 – present (gap May – Sept 1994)

10/28 – 07/63 SwifF = (TarbF – FishF – ResvF) * SwifXF / (SwifXF + TarbXG)
08/63 – 04/94 use actual daily values
05/94 – 09/94 SwifF = (TarbF – FishF – ResvF) * SwifXF / (SwifXF + TarbXG)
10/94 – 09/09 use actual daily values
Disaggregate filled in values with Fish

Little Fishing Creek FLOW (Lfsh), 10/59– present

10/28 – 09/59 LfshF = (FishF) * LfshXF / (LfshXF + FishXG)
10/59 – 09/09 use actual daily values
Disaggregate filled in values with Fish

Fishing Creek GAIN (Fish), 10/28– present

10/28 – 09/59 FishG = (FishF) * FishXG / (LfshXF + FishXG)
10/59 – 09/09 use actual daily values
Disaggregate all values with Fish

Tar River at Tarboro GAIN (Tarb), 10/31 – present (filled in back to 1928)

10/28 – 07/63 TarbG = (TarbF – FishF – ResvF) * TarbXG / (SwifXF + TarbXG)
02/71 – 09/73 TarbG = (TarbF - FishF - SwifF - CedrF - TarrF) * TarbXG / (LouiXG + ResvXG + TarbXG)
10/73 – 09/75 TarbG = (TarbF - FishF - SwifF - CedrF - LouiF - TarF) * TarbXG / (ResvXG + TarbXG)
10/75 – 12/78 TarbG = (TarbF - FishF - SwifF - LouiF - TarF) * TarbXG / (CedrXF + ResvXG + TarbXG)
Disaggregate whole record with Fish

Conetoe Creek near Bethel FLOW (Cone), 12/56 – 06/02

10/28 – 11/56 Use *fillin* but cannot scale because there are no downstream gage records
12/56 – 06/02 use actual daily values
07/02 – present ConeF = (GreeF – TarF) * ConeXF / (ConeXF + GreeXG)
Disaggregate filled in values using Fish

Tar River at Greenville GAIN (Gree), 04/97 – present

10/28 – 03/97 Use *fillin* but cannot scale because there are no downstream gage records
04/97 – 06/02 Use actual monthly gain
07/02 – present GreeG = (GreeF – TarF) * GreeXG / (ConeXF + GreeXG)
Disaggregate whole record with Fish

Section 4. Computing Inflows at OASIS Nodes from the Flows and Gains

This section describes the computation of inflows at OASIS nodes from flows and gains at gages described above. The computations are done "on the fly" during an OASIS run called "Compute_Inflows" using ratios of drainage area (total or incremental) as shown in Table 3. The computations are included in an OCL file called *set_inflows.ocl* and are listed below.

Node 010 (Tar R gage)	Tarr Flow
Node 050 (Louisburg gage)	Loui Gain
Node 60 (Cedar Ck)	CedrF *5 / 47.8
Node 66 (Taylor Ck)	CedrF *1 / 47.8
Node 070 (Cedar gage)	CedrF – Inflow60 – Inflow66
Node 120 (Reservoir)	Resv Gain
Node 190 (Little Fishing)	Lfsh Flow
Node 200 (Fishing Ck)	Fish Gain
Node 240 (Swift Ck)	Swif Flow
Node 300 (Tarboro gage)	Tarb Gain
Node 380 (Conetoe Ck)	Cone Flow
Node 390 (GUC WD)	GreeG * 2620 / 2660
Node 400 (Greenville gage)	GreeG – Inflow390

Table 3. List of Drainage Areas (in square miles)

Reservoirs			Gages		
Description	**Total Drainage Area**	**Incremental Drainage Area**	**Description**	**Total Drainage Area**	**Incremental Drainage Area**
Tar R. Reservoir	775	300.2	Tar River near Tar River	167	167
			Tar River at Louisburg	427	260
			Cedar Creek near Louisburg	47.8	47.8
			Swift Creek at Hilliardston	166	166
			Little Fishing	177	177
			Fishing Creek near Enfield	526	349
			Tar River at Tarboro	2,183	716
			Conetoe Creek near Bethel	78.1	78.1
			Tar River at Greenville	2,660	398.9

Section 5. Error Checking and Inflow Filtering

As noted in Section 3, because of the noisy data, a lot of error checking is necessary. These are some of the errors that can occur.

- Negative unimpaired gage flow. These are physically impossible and should be corrected.

- Negative gains. These are sometimes legitimate. However, there are times when a flood hits a gage at the very end of the month, while not arriving at the gage downstream until the beginning of the next month. This can cause a highly negative gain in the first month and a highly positive gain the next month. These should be corrected.

- There are pathological cases where the scaling can cause one gage to have a large positive flow, while the adjacent gage has a large negative flow. This can occur when the two extended values are similar in magnitude but opposite in sign. These need to be adjusted.

- For this work, the changes are tracked in the file ***hand_mods.xls*****.**

To test that all flows, gains, and drainage areas have been properly accounted for, we check that the "reconstituted," unimpaired gage flows match the actual unimpaired gage flows. In this case, reconstituted means that the flow at a given gage is computed by summing the appropriate upstream flows and gains. File ***gage_comp.xls*** compares the actual monthly unimpaired gage flows with the computed values from ***flows_gains_month.dss***. To test that all the drainage areas have been properly accounted for, the file ***gage_comp.xls*** compares the same actual unimpaired gage flows to the gage flows reconstituted from the computed OASIS inflows.

Monthly adjustments. Monthly flows were adjusted in ***hand_mods.xls***. Revisions were done for Tarr flows and Resv, Tarb and Gree gains in the ***unimpaired.dss*** file, and for the same flows/gains in file ***scale_flow_gain.dss***,to adjust for negatives from the scaling process as noted above.

To prevent model infeasibility from provisional inflows (see Appendix D), unrealistic releases from upstream reservoirs, or unrealistic water supply shortages, we added code in the OCL to filter these negative inflows. The negative inflow is "stored" until there is a sufficiently positive inflow to release the accumulated negative flows, thereby preserving mass over a multi-day period. Since the negative inflows are generally very small and infrequent, the filtering has negligible impact on being able to match the monthly unimpaired gage flow.

Section 6. Time of Travel / Flow Routing

To account for time of travel between points along the main stem of the Tar River, flow routing has been incorporated into both the development of inflows, and the handling of flows within the model. The time of travel (or lag) coefficients of 1-day in between reacheds were recommended by NC DWR. The time of travel reaches consist of the Tar River gage to the Louisburg gage, the Louisburg gage to the Tar River Reservoir, the Tar River Reservoir to the Tarboro gage, and the Tarboro gage to the Greenville gage. Flow routing for the development of inflows is handled in the HEC-DSSVue script called ***routing.py***. Flow routing within the model is handled in the OCL file ***routing.ocl***. The routing equations used for each gain are as follows:

Louisburg gain = Louisburg flow - yesterday's Tar River gage flow

Reservoir gain = Reservoir flow – yesterday's Louisburg gage flow

Tarboro gain = Tarboro flow – yesterday's Reservoir gage flow

Greenville gain = Greenville flow – yesterday's Tarboro gage flow

Note that the computation of gains may include subtracting the flows from tributaries in the reach (see Section 3), which are not routed.

When developing the inflows, the routing needs to be taken into account when computing gains.

To do this, first we create a daily routed time series for each flow; Tarr, Loui, Resv, and Tarb.

For example, this table illustrates the routed and actual unimpaired Tarr flows for five days in 2000:

Date	Tarr actual flow (cfs)	Tarr routed flow (cfs)
1/1/2000	44	46
1/2/2000	43	44
1/3/2000	41	43
1/4/2000	44	41
1/5/2000	267	44

Next, these daily routed timeseries are averaged monthly; the monthly flows are used to compute gains for the above mentioned reaches.

For example, in January 2000, the routed flow at Tarr is 176, and the flow at Louisburg is 501 cfs. The gain at Louisburg is computed by subtracting the routed Tarr flow from the Louisburg flow; 501 – 176 = 325 cfs.

Section 7. Extending the Record Beyond September 2009

As mentioned earlier, the finalized inflow record ends on September 30, 2009. This section describes how to finalize updates to the record when new records (including impairments) become available. This is not to be confused with *provisional* updates used to facilitate real-time forecasting, which are done directly from the model interface using the Update Record tab. Let us assume that we are adding data from October 1, 2009 to September 30, 2010. Note that we are only adding to the record. We are not changing any of the values prior to October 2009.

1. Assemble the new gage records in Table 4 and place in the files ***gage_day.dss*** and ***gage_month.dss***.

2. Compute the impairments at each gage and add them to the gage flows. This has been done in the unimpairment spreadsheet described earlier. Next put the daily unimpaired flows into the spreadsheet *unimpaired_summary.xls* to compute the monthly unimpaired flows using a spreadsheet pivot table.

3. Append the new unimpaired flows to the ***unimpaired.dss*** file (monthly averages) and to the ***unimpaired_daily.dss*** file (daily data) where appropriate.

4. Copy the following files into the folder *c:\Program Files\HEC\HecDssVue* (assuming the DSSVue has been installed in the default location).

    fillin.cf
    fillin.exe
    gage_day.dss
    gage_month.dss
    path_list.dat
    unimpaired.dss
    unimpaired_daily.dss

5. Copy the following files in folder of *c:\Program Files\HEC\HecDssVue\HecDssVue\scripts.* If extending the record to a year beyond 2009, be sure to change the ending years in the script from either '09 or 2009 to the appropriate year. Use the script files from the folder “scripts_2010”, in which this change has already been made.
    routing.py
    compute_gain.py
    convert_day_to_month.py
    convert_month_to_day.py
    disaggregate.py
    scale_flow_gain.py

6. Execute the script ***01_routing.py.*** This reads from file ***unimpaired_daily.dss***, creates routed time-series where necessary (see Section 6), and writes monthly routed flows to ***unimpaired.dss*** and ***extend_flow_gain.dss***.

7. Transfer the monthly flows from ***unimpaired.dss*** to *hand_mods.xls*. Here perform necessary modifications to remove negative gains from the records, following the examples of previous modifications. Then paste the modified flows and routed flows back into unimpaired.dss and ***extend_flow_gain.dss.***

8. Execute the script ***02_compute_gain.py.*** This reads from file ***unimpaired.dss***, computes all the gains, and writes to the files ***extend_flow_gain.dss*, *scale_flow_gain.dss*, and *fillin_input.dss.***

9. Close DSSVUE, and execute *fillin* by double-clicking it. This updates the file ***extend_flow_gain.dss*** with filled-in flows. Thus, this file is a combination of the actual unimpaired values and extended values.

10. Open up DSSVue again, execute the script ***03_scale_flow_gain.py***, which does the calculations described in Section 3. This reads from file ***extend_flow_gain.dss*** and updates file ***scale_flow_gain.dss***.

11. Import the appropriate flows and gains from ***scale_flow_gain.dss*** into *hand_mods.xls* and perform hand modifications to remove negative flows and gains. Paste modified flows/gains back into ***scale_flow_gain.dss***.

12. Execute the script ***04_convert_month_to_day.py***. This reads from files ***gage_month.dss*** and ***scale_flow_gain.dss*** to generate a daily record from the monthly records. The daily records are used to disaggregate the flows, as shown in Section 3. File ***month_day.dss*** is written.

13. Execute the script ***05_disaggregate.py*** to convert the monthly flows and gains at the gages to daily. File ***flow_gain.dss*** is produced; this file has the data that are read by OASIS.

14. Execute the script ***06_convert_day_to_month.py*** to convert the daily flows and gains at the gages into monthly values. The purpose of this step is to import the monthly flows and gains into ***gage_comp.xls*** to ensure that all the accounting has been properly done.

15. Once the correct accounting has been confirmed, place the ***flow_gains.dss*** file into the basedata folder, overwriting the original.

16. Execute the OASIS run called “Compute_Inflows”, beginning October 1, 2009 and ending September 30, 2010. This run contains a file called “set_inflows.ocl” that assigns inflows to the OASIS nodes.

    Then go to the run folder, open the ***ouput.dss*** file, click View → Refresh Catalog. Select all of the inflow nodes (i.e, those which have a pathname labeled inflow) and convert from acre feet (af) to cubic feet per second (cfs) by doing the following: click Utilities → Math Functions, in Arithmetic tab select Divide from the pull down menu, click the Data Set radio button and highlight all of the inflow records, click the Constant radio button and enter 1.9835 into the field, and then press the Compute button at the bottom of the window. Once the computation has completed, close the Math Functions window and click Yes when prompted to save the changes. Now select all of the inflow records and click Edit → Tabular Edit, and change all of the units to cfs, and all of the units Type to PER-AVER. Close the editing window and save your changes.

Next select all of the records, and copy and append to the ***basedata.dss*** file in the basedata folder. The basedata file will now contain this finalized data from October 1, 2009 to September 30, 2010. This basedata file will be used for all future runs.

Table 4. Gages Needed to Extend the Record Beyond September 2009

USGS Number	Description	Period of Record	Ref. Name	Drainage Area
02081500	TAR RIVER NEAR TAR RIVER, NC	10/1939-present	Tarr	167
02081747	TAR R AT US 401 AT LOUISBURG, NC	10/1973-present	Loui	427
02082506	TAR R BL TAR R RESERVOIR NR ROCKY MOUNT, NC	08/1972-present	Resv	777
02082770	SWIFT CREEK AT HILLIARDSTON, NC	08/1963-present	Swif	166
02082950	LITTLE FISHING CREEK NEAR WHITE OAK, NC	10/1959-present	Lfsh	177
02083000	FISHING CREEK NEAR ENFIELD, NC	10/1926-present	Fish	526
02083500	TAR RIVER AT TARBORO, NC	10/1931-present	Tarb	2,183
02084000	TAR RIVER AT GREENVILLE, NC	04/1997-present	Gree	2,660