4.2.7. Summarize all unique pumpage totals by grid cell id.

4.2.7.1. Summarize all the "pump_un*yy*" fields by grid cell id, by using the summarize button and adding "pmp_80" (sum) through "pmp_99" (sum) in the dialog box. Name this summary file ***area*_irr_pumpbygrid_80_99**. (i.e. **sw_irr_pumpbygrid_80_99.dbf**).

4.2.8. Vertical Distribution: Follow procedures outlined in sections 4.5.

4.2.9. Import irrigation pumpage table back into MS Access database as a table ***area*_irrigation_total**, e.g., **sw_irrigation_total**

4.2.9.1.In MS Access, import the attribute table for the Arcview shape file **grid_irr_*yy*.dbf** as a dbase file. This table should include one record for each possible Grid_ID, and at least the fields "Grid_ID", "year", and "pump*yy*_IRR."

4.2.10. The table ***area*_irrigation_total** now has only the grid_id of the upper model, i.e., the first digit is 1. The actual vertical distribution data is in the fields "per1" to "per*x*" where x is the number of vertical layers in the model. Copy the table x-1 times in an append query, incrementing the first digit of the grid id, to create a record for each model layer. There now should be L times the original number of records in the table. For example, for the northwestmost grid cells of a model with four layers, the following grid id's should now exist: 1001001, 2001001, 3001001, and 4001001; whereas only 1001001 was in the original table.

4.2.11. Calculate for each year the actual pumpage for each record as the product of the pumpage for a given year multiplied by the percent of pumpage from that model layer (from the fields "per1" – "per4", for a model with 4 layers).

4.2.12. Create a new summary query **Irrigation_annual_*area*** to summarize the pumpage for each grid_id and year from the table ***area*_irrigation_total**.

4.3. Spatial allocation of livestock groundwater pumpage – Livestock groundwater use within each county-basin is distributed evenly to all rangeland, Anderson Level II land use codes 31 (herbaceous rangeland), 32 (shrub and brush rangeland), and 33 (mixed rangeland) of the USGS 1:250,000 land use land cover data set (http://edcwww.cr.usgs.gov/glis/hyper/guide/1_250_lulc).

4.3.1. Determine rangeland within each county-basin

4.3.1.1.In Arcview, create a rangeland-only land use shapefile by loading the USGS land use shapefiles by quadrangle, merging them as required to cover the model domain, selecting the land use codes 31, 32, and 33 in a query, then saving the theme as a new shapefile **Rangeland.shp**.

4.3.1.2.Using the Geoprocessing Wizard, intersect the Rangeland shapefile with the County-basin shapefile (make sure to use entire county basin areas, and not the "clipped to domain" version) to make a new intersection shapefile **range_countybasin.shp**.

4.3.1.3.Calculate the unique area (in square miles) of the new intersected polygons "area_un1" using the field calculator (area_un1=shape.returnarea/27878400).

4.3.1.4.Summarize the unique area by county-basin (total area of rangeland within county-basin) using the summary button.

4.3.1.5.Link the summary table back to the range_countybasin shape file and migrate it into a new field "rg_cb_tot" using the field calculator.

4.3.1.6.Determine weighted area factor "w_area1" for each polygon using the field calculator (w_area1)=(area_un1 / rg_cb_tot). W_area1 is, for each rangeland polygon, the fraction of the total rangeland area within the county-basin.

4.3.2. Intersect the rangeland/countybasin polygons with the Model Grid and set up for unique pumpage calculations.

4.3.2.1. Using the Geoprocessing Wizard, intersect the shapefiles range_countybasin and Model Grid to create a new shape file **rng_cb_mg.shp**.

4.3.2.2. Calculate the unique area of "intersected" polygons (area_un_grid) using the field calculator (area_un_grid=shape.returnarea/27878400). Double check that no values are greater that 1.

4.3.2.3. Determine the weighted area factor (w_area_grid) = (area_un_grid/area_un1).

4.3.3. Calculate unique pumpage "pump_un_yy" for the intersected polygons for every year (80-99).

4.3.3.1. Add the fields "pump_un80" – "pump_un99" to the **rng_cb_mg** attribute table.

4.3.3.2.Using SQL Connect, query the Access table **PumpagebyMajorAquifer1980-1999** for all years.

4.3.3.3. Query the records (by the year column) for each year, and specific aquifer (by aquifer code column) and export each query as a separate .dbf file. "Pump_by_cb_*yyyy_aquifer*.dbf." These tables will have a column for each use category, and can therefore be used in the irrigation calculations for the same aquifer of concern.

4.3.3.4. Join the table "pump_by_cb_1980.dbf" to the attribute table "rng_cb_mg" by countybasin. (make certain that all countybasin names are spelled the same).

4.3.3.5. Calculate "pump_un80" using the field Calculator (pump_un80 = w_area_grid * (w_area_1 * livestock)). (livestock is the column of the joined table "pump_by_cb_1980" that contains the countybasin annual pumpage totals for livestock use).

4.3.3.6. Repeat 4.3.3.4 – 4.3.3.5 for all years.

4.3.4. Summarize all unique pumpage totals by grid cell id.

4.3.4.1. Summarize all the "pump_un*yy*" fields by grid cell id, by using the summarize button and adding "pump_un_80" (sum) through "pump_un_99" (sum) in the dialog box. Name this summary file "*area*_stk_pumpbygrid_80_99." (i.e. sw_stk_pumpbygrid_80_90.dbf).

4.3.5. Vertical Distribution: Follow procedures outlined in sections 4.5.

4.3.6. Import livestock pumpage summary table back into MS Access database as a table **area_livestock_total**, e.g, **sw_livestock_total**.

4.3.7. The table ***area*_livestock_total** now has only the grid_id of the upper model, i.e., the first digit is 1. The actual vertical distribution data is in the fields "per1" to "perx" where x is the number of vertical layers in the model. Copy the table x-1 times in an append query, incrementing the first digit of the grid id, to create a record for each model layer. There now should be L times the original number of records in the table. For example, for the northwestmost grid cells of a model with four layers, the following grid id's should now exist: 1001001, 2001001, 3001001, and 4001001; whereas only 1001001 was in the original table.

4.3.8. Calculate for each year the actual pumpage for each record as the product of the pumpage for a given year multiplied by the percent of pumpage from that model layer (from the fields "per1" – "per4", for a model with 4 layers).

4.3.9. Create a new summary query **Livestock_annual_*area*** to summarize the pumpage for each grid_id and year from the table ***area*_irrigation_total**.

4.4. Spatial allocation of rural domestic (C-O) groundwater pumpage.

4.4.1. Calculate the Population in each 1 mile grid cell.

4.4.1.1. In Arcview, load the 1990 block-level census population shapefile.

4.4.1.2. Load Arcview polygon shapefiles for cities. Select census blocks that fall with in city boundaries and delete those records so that rural domestic pumpage does not get distributed to cities. (Note: we're assuming that city boundaries are good surrogates for the extent of the area served by public water supply systems, whose pumpage is reported under the category "MUN").

Repeat this process for the reservoir areas.

4.4.1.3.Calculate the area of census blocks in sq. miles in a new field "blk_area" using the Field Calculator function (blk_area=shape.returnarea / 27878400).

4.4.1.4. Load the model grid, model domain, and county-basins shapefile. Select all county-basins that are intersected by the model domain boundary. Union the selected county-basins with the model domain boundary. In the resulting shapefile, delete the polygons that are inside the model domain, leaving only areas of the county-basins that are outside of the model domain. Dissolve these polygons into one and merge with the model grid shapefile. Give this new record a grid_id of 9999999. (Adding this new area will insure that, when the county-basin total populations are calculated, the population outside of the model domain will be included).

4.4.1.5. In the Geoprocessing Wizard, intersect the census block shapefile with the model grid shapefile to create a new shape file **intrsct*90*.shp**. (Note: Because the model grid size is 1 square mile, no intersected polygon (inside the model domain) should be larger than 1 square mile. Make sure that this is the case before proceeding).

4.4.1.6. Calculate the unique area of all intersected polygons in square miles as a new field "area_un1" using the Field Calculator function (area_un1=shape.returnarea / 27878400). (so that one grid cell has an area of 1).

4.4.1.7. Add a new numeric field "pop_un1" – the unique Population of the intersected polygons. Using the Field Calculator, calculate its value as (POP_un1 = pop*90* * area_un1 / blk_area) where pop*90* is the block Population from the census file.

4.4.1.8. Sum the field "pop_un1" by grid_id using the Field Summarize function to calculate the total population within each grid cell. Join this summary table to the original grid table by grid_id and copy value into new field "pop_90".

4.4.1.9. Repeat steps 4.5.1.1 – 4.5.1.8 (no need to repeat step 4.5.1.4, just use the grid file that was used for previous iteration).

4.4.2. Calculate the rural domestic pumpage for each 1 mile grid cell.

4.4.2.1. Intersect the county-basins shapefile with the model grid (which now has census populations for 1990 and 2000) to create a new shapefile **grid_cb_pop**.

4.4.2.2. Create new field "area_un2" and calculate unique area using field calculator ("area_un2" = [shape].returnarea/27878400)

4.4.2.3. Create two new fields "pop_un90" and "pop_un00". Calculate using the field calculator ("pop_un*yy*" = "area_un2"/ "pop_*yy*")

4.4.2.4. Using SQL Connect, query the Access table **PumpagebyMajorAquifer1980-1999** for all years.

4.4.2.5. Query the records (by the year column) for each year (because Rural Domestic pumpage data is not aquifer specific, there is no need to query by aquifer) and export each query as a separate .dbf file. "Pump_by_cb_*yyyy*.dbf."

4.4.2.6. Join table "pump_by_cb_1980.dbf" to grid_cb_pop.dbf by county-basin.

4.4.2.7. Add field "pmp80." Using field calculator, calculate "pmp80" (pmp80=CO*pop_un90/cb_pop90).

4.4.2.8. Repeat steps 4.5.2.6 – 4.5.2.7 for each year. Use pop90 for years 1980-1989 and use pop00 for years 1990-1999.

4.4.2.9. As a quality control check, sum the values of "rdom_pump" for each county-basin and make sure it matches the total for the county-basin from the Access table.

4.4.2.10. Summarize pmp80 through pmp99 by grid id. Link summary back to model grid file and migrate pumpage values.

4.4.3. Vertical Distribution: Follow procedures outlined in section 4.5.

4.4.4. Import the rural domestic pumpage table into the MS Access database as a table ***area*_rurdom_total**, e.g., **sw_rurdom_total**.

4.4.5. The table ***area*_rurdom_total** now has only the grid_id of the upper model, i.e., the first digit is 1. The actual vertical distribution data is in the fields "per1" to "per*x*" where x is the number of vertical layers in the model. Copy the table x-1 times in an append query, incrementing the first digit of the grid id, to create a record for each model layer. There now should be L times the original number of records in the table. For example, for the northwestmost grid cells of a model with four layers, the following grid id's should now exist: 1001001, 2001001, 3001001, and 4001001; whereas only 1001001 was in the original table.

4.4.6. Calculate for each year the actual pumpage for each record as the product of the pumpage for a given year multiplied by the percent of pumpage from that model layer (from the fields "per1" – "per4", for a model with 4 layers).

4.4.7. Create a new summary query **Rurdom_annual_*area*** to summarize the pumpage for each grid_id and year from the table ***area*_rurdom_total**.

4.5. Vertical Distribution of groundwater pumpage. *Note: These procedures are for all use categories, and this section is referenced multiple times. Take care, and perform only the operations that apply to that particular use.

4.5.1. Assign default well depths to model grid cells – Most, but not all, well-specific pumpage from the categories MUN, MFG, PWR, and MIN are associated with a reported well depth, screened interval, land surface elevation, which are used to attribute the pumpage to a specific vertical model layer. For those wells whose depth, screened interval, or land surface elevation is unknown, and for the non-well-specific pumpage in the categories C-O, STK, and IRR, it is necessary to interpolate these depths/elevations to assign the pumpage to a specific model layer. In this procedure, the approach is to interpolate on the basis of the depths of nearby (<10 miles) wells. On average, municipal, industrial, and irrigation water wells tend to be deeper than rural domestic or livestock wells. Thus, if there are nearby wells in the same water use category, the interpolation is based on these wells. In the absence of nearby wells of the same use category, the interpolation is based on nearby wells of any water use category. **The procedures outlined in section 4.5.1 cover all use categories, and therefore, only need to be done once per model area.*

4.5.1.1. In Arcview, using SQL Connect, query the MS Access database table **All_wells** for all wells in the major aquifer of concern (based on the field "aqfr_id_1"). Save this query as a table ***AQ*_wells**, where ***AQ*** is a 2-character code representing the aquifer of interest.

4.5.1.2. Load these wells in a View as an event theme, using the fields lat_dd as y-coordinate and long_dd as x-coordinate. Convert the event theme to GAM projection as per GAM Technical Memo 1-01, then save this theme as a shape file.

4.5.1.3. Query the shape file's attribute table for all domestic water wells (water_use_1 = "domestic").

4.5.1.4. Using Arcview Spatial Analyst, under the Analyst, Properties menu, set analysis extent and grid size to be equal to the GAM model grid.

4.5.1.5. Next, under the Surface menu, interpolate a grid with values of interpolated well depth, via the inverse distance weighting method, within a fixed radius of 10 miles, with a power of 2.

4.5.1.6. Repeat steps 4.5.1.3 – 4.5.1.5 to create an interpolated well depth grid for each of the other water use categories MUN, MFG, PWR, MIN, STK, and IRR, as well as a well depth grid for all water use categories combined.

4.5.1.7. When a depth was not reported for a well, these grid values can be used as an estimated well depth. A new text field "depth source" is added to the well table to indicate that the well depth was estimated by interpolation, not reported. This allows a hydrogeologist or modeler to review these wells to make sure they fall in the proper model layer. When a well depth is checked and corrected manually, a value of "manual" is entered in the field "depth source'. Valid values of depth source include "reported", "interpolated", or "manual".

4.5.2. Assign default screened intervals to wells – For wells with no reported screened interval, calculate the well screened interval. The lower boundary is the well depth, while the upper boundary of the screened interval is calculated as the well depth minus an estimated screen length. The default screen lengths will be estimated from other wells in the same aquifer for which the screened interval is known.

4.5.2.1. An Excel file ***Screened_Interval.xls*** is provided by the modelers. It contains the land surface elevation and depths to the top and bottom of the screen for each well. The screened interval is calculated as the difference between the top and bottom depths. This file is loaded in Arcview and joined to the ***AQ_Wells*** table by state well number. Next, under the Surface menu, interpolate a grid with values of interpolated screened interval, via the inverse distance weighting method, within a fixed radius of 10 miles, with a power of 2.

4.5.2.2. When a screened interval is not reported for a well, these grid values can be used to estimate the upper depth of the screened interval, assuming that the well depth is the bottom of the interval. A new text field "screen_source" is added to the well table to indicate that the well depth was estimated by interpolation, not reported. Valid values of screen source include "reported" or "interpolated", or "manual".

4.5.3. Assign land surface elevations to wells – For wells without a reported land surface elevation (in the field "elev of lsd") a land surface elevation must be estimated. For this purpose, a 30-meter digital elevation model (DEM) grid is added to an Arcview project with the well data table. The Arcview script "getgridvalue" in Appendix 2 is run to return the value of the land surface elevation for the well.

4.5.4. Estimate the screened interval for non-well-specific pumpage - For the non-well-specific uses STK, IRR, and C-O, in order to distribute the pumpage vertically, each model grid cell may be treated as a well. Using the centroids of the model grid cells as if they were wells, copy the interpolated values of well depth, screened interval, and land surface elevation to each grid cell as described above.

4.5.5. Convert depths to elevations - In order to compare to model layers, which are reported as elevation (feet above mean sea level), it is necessary to convert the depths of the top and bottom of screened intervals to elevations. To do this, subtract the depths from the land surface elevation, in feet above mean sea level.

4.5.6. Determine vertical distribution of pumpage totals by comparing the elevations of the top and bottom of the well screened interval to model layer elevations. (For point source water use categories, this will be done for each specific well. For non-point source this will be done for each 1 mile grid cell).

4.5.7. Spatially join the flow layer structure (model grid cells with tops of aquifer elevations) to the wells. (for non-point source join by grid id).

4.5.8. Run vertical distribution avenue script on points (see appendix for code). This script will place a "pumpage percentage" in the flow layer percentage columns (per1 – per6). This value is actually the percentage of the total length of the screened interval that resides in each flow layer (possible 0 – 100).

4.5.9. Once script is successfully run, a series of QA checks must be run, and in certain cases percentage values must be altered manually. Field "calc_code" will be given a specific code for each case of manual alteration.

4.5.9.1. Query records that have a value of "99999" for every layer elevation (i.e. layer doesn't exist at that location). Set calc_code to "N".

4.5.9.2. Query records whose top of screen elevation is shallower than the top of the shallowest existing layer. (i.e. (top of layer 2 = 999999 and per2 > 0)). The script automatically puts a value in per2 if the top of screen is shallower than layer 3, but if layer 2 doesn't exist there then per2 should be zero and the value should be shifted down. In this case, calc_code should be set to "S3". This will tell someone that the screen is shallower than the shallowest layer which is layer 3.

4.5.9.3. Query records whose depth is deeper than the bottom layer. (i.e. depth<bottom layer). Put the remainder of the pumpage that was lost below into the bottom layer and set calc_code to "D".

4.5.9.4. Query records whose screened interval spans layer 1 or 2 and enters layer 3 (Carrizo). (i.e. per3>0 and per2>0). It is assumed that if the screened interval reaches the Carrizo then all of the water is being taken from that layer and not the above layers of inferior quality. Set per1 and per2 to zero and add their values to per3. Set calc_code to "C".

4.5.9.5. Query records whose reported top of screen elevation is less than the bottom of screen elevation. Manually set the appropriate layer percentage to 100%. Set calc_code to "E".

4.5.9.6. Query records whose top of screen elevation exactly equals one of the layer top elevations. This is very rare, but if it happens, the percentage value must be manually entered. Set calc_code to "=".

4.5.9.7. Query records whose total percentage is less than 100% by less than .5%. Due to a program glitch values of 99.5% get rounded to 100% and the rest is left out. Manually set percentage value to 100%. Set calc_code to "R".

4.5.9.8. Query all other records (records that don't have a calc_code value and whose tot_per = 100%). Set calc_code to "NP" for no problems.

5. Temporal Distribution of Rural Domestic, Livestock, and Irrigation Groundwater Use

5.1. Temporal distribution of livestock pumpage - Because we have only annual total groundwater pumpage estimates for STK, we need to derive monthly pumpage estimates. According to TWDB GAM Technical Memo 01-06, annual total livestock pumpage may be distributed uniformly to months since the water needs of livestock are not likely to vary significantly over the course of a year.

5.1.1. In the MS Access database, create a new table called Monthly Factors with the fields "countyname", "basinname", "countynumber", "basinnumber", "data_cat", "year", and "month". The table should include a record for every county-basin within the model domain, water use category "data_cat", year (1980-1999), and month (1-12), as well as an additional annual total record (month="0") for each county-basin, year, and water use category. Add 2 new fields "mfraction" and "Monthly distribution factor source" to the new table. The former is the numeric monthly distribution factor, while the latter is a text field indicating the source of the distribution factor. For all monthly livestock water use records (data_cat=STK, month in 1-12), enter an mfactor of "0.0833" (1/12) and a monthly distribution factor source of "Tech Memo 01-06". For all annual total water use records (data_cat=STK, month =0), enter an mfactor of "1" and a monthly distribution factor source of "NA".

5.2. Temporal distribution of irrigation (IRR) pumpage - Because we have only annual total groundwater pumpage estimates for IRR, we need to derive monthly pumpage estimates. Monthly distribution factors will be derived separately for rice-farming counties and non-rice-farming counties.

5.2.1. Temporal distribution of groundwater used for non-rice irrigation –

5.2.1.1.Record monthly crop evapotranspiration (ET), or total water demand, for each of the Texas Crop Reporting Districts (TCRDs) that occur within the model domain, from the report "Mean Crop Consumptive Use and Free-Water Evaporation for Texas" by J. Borrelli, C.B. Fedler, and J.M. Gregory, Feb. 1, 1998 (TWDB Grant No. 95-483-137). Use these values for all years.

5.2.1.2.Next, determine monthly precipitation (P) for the period 1980-1999 for the locale within each of the TCRDs that occur within the model domain.

5.2.1.3.Determine the monthly water deficit for each month of the two periods 1980-1989 and 1990-1999 by subtracting the P values from the ET values for each TCRD. Replace negative values with zero. Sum all water deficit values by month for each of the two periods, and divide by the number of months in each period to obtain an average non-rice monthly distribution factor for each month for the two periods 1980-89 and 1990-99.

5.2.2. Temporal distribution of groundwater used for rice irrigation –

5.2.2.1.First, identify the counties within the model area where rice is irrigated, using the 1989 and 1994 irrigation reports. Include only those counties in this analysis.

5.2.2.2.Next, using monthly pump power usage records provided by rice farmers, calculate monthly distribution factors for total annual power usage. Average all distribution factors within a county to get an average rice irrigation distribution factor.

5.2.3. Develop composite irrigation monthly distribution factors for each county and year based on the monthly factors for rice and non-rice irrigation, and the fraction of irrigation for rice in that county.

5.2.3.1.The TWDB irrigation survey data files Irr1989.xls and Irr1994.xls contain reported irrigation water use estimates for each crop and county. From these tables, calculate the fraction of irrigation water for rice in each county for the 1980s (based on 1989) and the 1990's (based on 1994).

5.2.3.2.Calculate the composite monthly distribution factor (MF_{comp}) for irrigation for each county as:

$$MF_{comp} = MF_{rice} * X + MF_{non\text{-}rice} * (1 - X)$$

where X is the fraction of water used for rice, and MF_{rice} and $MF_{non\text{-}rice}$ are the monthly distribution factors for rice and non-rice crops determined in steps 5.2.1 and 5.2.2, above.

5.2.4. For the county-basins where rice is not irrigated, enter the monthly distribution factors from step 5.2.3, above, in the table **Monthly Factors** for each year, county, basin, using "data_cat"="IRR", and "Monthly Distribution Factor Source"="ET/P Water Deficit Analysis."

5.2.5. For the county-basins where rice is irrigated, enter the monthly distribution factors from step 5.2.3, above, in the table **Monthly Factors** for each year, county, basin, using "data_cat"="IRR", and "Monthly Distribution Factor Source"="ET/P + Power Usage Analysis."

5.3. Temporal distribution of rural domestic (C-O) pumpage - Because we have only annual total groundwater pumpage estimates for C-O, we need to derive monthly pumpage estimates. According to TWDB GAM Technical Memo 01-06, annual rural domestic pumpage may be distributed based on the average monthly distribution of all municipal water use within the same county-basin.

5.3.1. In a MS Access query based on the table **RawDataMUN_linkedwithwellinfo**, calculate the sum of the fields "Annual total in gallons", "jan", "feb",.....,"dec" for each county, basin, and year.

5.3.2. Next, calculate "mfraction," the fraction of the annual total for each month, by dividing the columns "sum of jan", "sum of feb",....,"sum of dec" by the "sum of annual total in gallons.". Transpose this table via a query to make a table with the following fields: "countyname", "basinname", "year", "month", "mfraction", "data_cat," and "monthly distribution factor source." A value of "C-O" should be entered in the field "data_cat", and the value of "monthly distribution factor source"="this county-basin mun."

5.3.3. The values of "mfraction" are statistically reviewed for outliers. Generally, monthly distribution factors fall within the range 0.035 to 0.15. Higher or lower values can be found when there is little municipal water use in a county-basin. In this case, substitute the values of "mfraction" from an adjacent county-basin, preferably from within the same county. Update the field "monthly distribution factor source" with the name of the county-basin used as a source.

5.3.4. For Louisiana and Arkansas parishes and counties, use the monthly distribution factors of the nearest Texas county-basin.

5.3.5. Add an annual total record for each county-basin-year, with "data_cat"="C-O", "month"="0", "mfraction"="1", and "monthly distribution factor source"="NA."

5.3.6. Using an append query, append these records to the table **Monthly Factors**.

6. Summarize Pumpage Information

6.1. Summary Queries

6.1.1. Queries for livestock - Create a new select query **MMMYY_STK** to calculate pumpage for the month and year of interest by multiplying the monthly factor for that month, year, and water use category, in the table **Monthly Factors**, by each entry in the imported table **Livestock_annual_CGC**. For any specified month (MMM) and year(YY), the SQL for the query **MMMYY_STK** is:

```
SELECT Livestock_annual_CGC.GRID_ID, Livestock_annual_CGC.DATA_CAT,
Livestock_annual_CGC.Year, Livestock_annual_CGC.MODEL, [MONTHLY
FACTORS].MONTH, [SumPumpageAF]*[mfraction] AS PumpageAF

FROM Livestock_annual_CGC LEFT JOIN [MONTHLY FACTORS] ON
(Livestock_annual_CGC.Year = [MONTHLY FACTORS].YEAR) AND
(Livestock_annual_CGC.DATA_CAT = [MONTHLY FACTORS].DATA_CAT)
AND (Livestock_annual_CGC.basinnum = [MONTHLY FACTORS].basinnum)
AND (Livestock_annual_CGC.CountyNumber = [MONTHLY
FACTORS].countynum)

WHERE (((Livestock_annual_CGC.DATA_CAT)="STK") AND
((Livestock_annual_CGC.Year)=1980) AND
((Livestock_annual_CGC.MODEL)="CGC") AND (([MONTHLY
FACTORS].MONTH)=1))

ORDER BY [SumPumpageAF]*[mfraction];
```

6.1.2. Queries for irrigation – Create a new select query **MMMYY_IRR** to calculate pumpage for the month and year of interest by multiplying the monthly factor for that month, year, and water use category, in the table **Monthly Factors**, by each entry in the imported table **Irrigation_annual_CGC.** For any specified month (MMM) and year(YY), the SQL for the query **MMMYY_IRR** is:

SELECT Irrigation_annual_CGC.GRID_ID, Irrigation_annual_CGC.DATA_CAT, Irrigation_annual_CGC.Year, Irrigation_annual_CGC.MODEL, [MONTHLY FACTORS].MONTH, [SumPumpageAF]*[mfraction] AS PumpageAF

FROM Irrigation_annual_CGC LEFT JOIN [MONTHLY FACTORS] ON (Irrigation_annual_CGC.basinnum = [MONTHLY FACTORS].basinnum) AND (Irrigation_annual_CGC.CountyNumber = [MONTHLY FACTORS].countynum) AND (Irrigation_annual_CGC.Year = [MONTHLY FACTORS].YEAR) AND (Irrigation_annual_CGC.DATA_CAT = [MONTHLY FACTORS].DATA_CAT)

WHERE (((Irrigation_annual_CGC.DATA_CAT)="IRR") AND ((Irrigation_annual_CGC.Year)=1980) AND ((Irrigation_annual_CGC.MODEL)="CGC") AND (([MONTHLY FACTORS].MONTH)=1))

ORDER BY [SumPumpageAF]*[mfraction];

6.1.3. Queries to summarize rural domestic (county-other) - Create a new select query **MMMYY_C-O** to calculate pumpage for the month and year of interest by multiplying the monthly factor for that month, year, and water use category, in the table **Monthly Factors**, by each entry in the imported table **Rurdom_annual_CGC**. For any selected month (MMM) and year(YY), the SQL for the query **MMMYY_C-O** is:

SELECT Rurdom_annual_CGC.GRID_ID, Rurdom_annual_CGC.DATA_CAT, Rurdom_annual_CGC.Year, Rurdom_annual_CGC.MODEL, [MONTHLY FACTORS].MONTH, [SumPumpageAF]*[mfraction] AS PumpageAF

FROM Rurdom_annual_CGC LEFT JOIN [MONTHLY FACTORS] ON (Rurdom_annual_CGC.DATA_CAT = [MONTHLY FACTORS].DATA_CAT) AND (Rurdom_annual_CGC.Year = [MONTHLY FACTORS].YEAR) AND (Rurdom_annual_CGC.CountyNumber = [MONTHLY FACTORS].countynum) AND (Rurdom_annual_CGC.basinnum = [MONTHLY FACTORS].basinnum)

WHERE (((Rurdom_annual_CGC.DATA_CAT)="C-O") AND ((Rurdom_annual_CGC.Year)=1980) AND ((Rurdom_annual_CGC.MODEL)="CGC") AND (([MONTHLY FACTORS].MONTH)=1))

ORDER BY [SumPumpageAF]*[mfraction];

6.1.4. Query to summarize well-specific pumpage - Create a new select query in MS Access **MMMYYWell-SpecificSum** to summarize the well-specific pumpage from all wells within a grid cell for the desired month or year. For any specified month and year, the SQL query for well-specific pumpage would be:

```
SELECT CGC_gridsum_well_specific.GRID_ID, "WS" AS DATA_CAT,
CGC_gridsum_well_specific.year, CGC_gridsum_well_specific.Model,
CGC_gridsum_well_specific.month,
CGC_gridsum_well_specific.SumPumpage_af AS PumpageAF

FROM CGC_gridsum_well_specific

WHERE (((CGC_gridsum_well_specific.year)=[Enter year]) AND
((CGC_gridsum_well_specific.Model)="CGC") AND
((CGC_gridsum_well_specific.month)=[Enter month]))

ORDER BY CGC_gridsum_well_specific.SumPumpage_af;
```

6.1.5. In order to ensure that each grid cell is included in the final summary queries, even if there is no pumpage from the cell, we must create a full grid with values of zero.

6.1.5.1.Create a new table **Zero_grid_annual** in a make-table query based on the table **grid_lkup_*area*** with one record for each grid cell and year. For instance, a model with 212 rows, 180 columns, and 6 layers, for 20 years would be create a table with 212 x 180 x 6 x 20= 4,579,200 records. In the make-table query, add a field "SumPumpageAF" with a value of zero for each record.

6.1.5.2.Create a new query **MMMYY_ZeroGrid** to provide zero values for each grid cell for each month. You can use any of the monthly factors, as all results will equal zero. As an example, the SQL query for January 1980 would be:

```
SELECT Zero_Grid_Annual.GRID_ID, Zero_Grid_Annual.DATA_CAT,
Zero_Grid_Annual.Year, Zero_Grid_Annual.MODEL, [MONTHLY
FACTORS].MONTH, Zero_Grid_Annual.SumPumpageAF

FROM Zero_Grid_Annual LEFT JOIN [MONTHLY FACTORS] ON
(Zero_Grid_Annual.basinnum = [MONTHLY FACTORS].basinnum) AND
(Zero_Grid_Annual.CountyNumber = [MONTHLY FACTORS].countynum)
AND (Zero_Grid_Annual.Year = [MONTHLY FACTORS].YEAR)

WHERE (((Zero_Grid_Annual.Year)=[Enter year]) AND (([MONTHLY
FACTORS].MONTH)=[Enter month]) AND (([MONTHLY
FACTORS].DATA_CAT)="IRR"))

ORDER BY Zero_Grid_Annual.GRID_ID;
```

6.1.6. In Access, create a new union query **MMMYYUnionofPumpage** to combine the domestic, livestock, rural domestic, and well-specific pumpage sums, as well as the zero value, for each grid cell. As an example, the SQL for any given year and month is:

```
SELECT * FROM [MMMYY_C-O] UNION ALL SELECT * FROM [MMMYY_IRR] UNION ALL SELECT * FROM [MMMYY_STK] UNION ALL SELECT * FROM [MMMYY_ZeroGrid] UNION ALL SELECT * FROM [MMMYYWell-specificSum];
```

6.1.7. Create a new select query **SumPumpageGrid_MMMYY** to summarize all pumpage by grid cell, grouping by grid_id, month, and year the pumpage from the above union query. As an example, the SQL for January 1980 is:

```
SELECT MMMYYUnionofPumpage.GRID_ID,
MMMYYUnionofPumpage.Year, MMMYYUnionofPumpage.MONTH,
Sum(MMMYYUnionofPumpage.PumpageAF) AS SumOfPumpageAF,
Sum([PumpageAF]*[MGDfromAF]) AS PumpageMGD

FROM MMMYYUnionofPumpage LEFT JOIN UnitConversion ON
MMMYYUnionofPumpage.MONTH = UnitConversion.Month

GROUP BY MMMYYUnionofPumpage.GRID_ID,
MMMYYUnionofPumpage.Year, MMMYYUnionofPumpage.MONTH

ORDER BY MMMYYUnionofPumpage.GRID_ID;
```

6.2. Join pumpage queries to Arcview shapefile if visual display of the results for a month or year is desired.

6.2.1. In Arcview, import the MS Access query **SumPumpageGrid_MMMYY**, and join it to the model grid cells in the Arcview shapefile based on the field "Grid_ID."

6.2.2. In Arcview, import the MS Access queries **MMMYY_STK, MMMYY_IRR, MMMYY_C-O**, and **Well-specificpumpage.** Link these tables to the model grid cells in the Arcview shapefile based on the field "Grid_ID" and, for well-specific pumpage, "year." Selection of a grid cell in Arcview will then also select the records in each of these tables that pump from the grid cell selected.

Appendix 1 - Vertical Distribution Avenue Script

```
theView = Av.GetActiveDoc
theTheme = theView.findTheme("wells")
theFtab = theTheme.GetFtab

'get elevation values for layers
theLay1Field = theFtab.findField("top_younge")
theLay2Field = theFtab.findField("top_reklaw")
theLay3Field = theFtab.findField("top_carriz")
theLay4Field = theFtab.findField("top_uwilco")
theLay5Field = theFtab.findField("top_mwilco")
theLay6Field = theFtab.findField("top_lwilco")
theBottomField = theFtab.findField("bas_lwilco")

'get percentfield holders
thePer1Field = theFtab.findField("per1")
thePer2Field = theFtab.findField("per2")
thePer3Field = theFtab.findField("per3")
thePer4Field = theFtab.findField("per4")
thePer5Field = theFtab.findField("per5")
thePer6Field = theFtab.findField("per6")
theTotPerField = theFtab.findField("tot_per")

'get well values
theScreenField  = theFtab.findField("Screen")
theDepthField  = theFtab.findField("depth")

theSel = theFtab.GetSelection

for each rec in theSel
 ct = 0
 totPerVal = 0
 cumPerVal = 0
  theDepthVal = theFtab.ReturnValue(theDepthfield,rec)
  theScreenVal = theFtab.ReturnValue(theScreenfield,rec)
  screenLengthVal = (theScreenVal - theDepthVal).abs

  theLay1Val = theFtab.ReturnValue(theLay1field,rec)
  theLay2Val = theFtab.ReturnValue(theLay2field,rec)
  theLay3Val = theFtab.ReturnValue(theLay3field,rec)
  theLay4Val = theFtab.ReturnValue(theLay4field,rec)
  theLay5Val = theFtab.ReturnValue(theLay5field,rec)
  theLay6Val = theFtab.ReturnValue(theLay6field,rec)
  theBotVal =  theFtab.ReturnValue(theBottomField,rec)

  if ((theScreenVal < theLay1Val ) And (theScreenVal > theLay2Val)) then
    if (theDepthVal < theLay2Val) then
      per1 =  (((theLay2Val - theScreenVal) / screenLengthVal) * 100).abs
      theFtab.SetValue(thePer1field,rec,per1)
      cumPerVal = cumPerVal + per1
    else
      per1 = (100 - cumPerVal)
      cumPerVal = cumPerVal + per1
```

```
      theFtab.SetValue(thePer1field,rec,per1)
    end
  else
      per1 = 0
      theFtab.SetValue(thePer1field,rec,per1)
  end
'---------------------------------------------layer 2
  if (cumperval.round = 100) then
     'continue
     ct=ct+1
     per2 = 0
     theFtab.SetValue(thePer2field,rec,per2)
  else
   if ((theScreenVal < theLay2Val ) And (theScreenVal > theLay3Val)) then
     if (theDepthVal < theLay3Val) then
       per2 =  (((theScreenVal - theLay3Val) / screenLengthVal) * 100).abs
       cumPerVal = cumPerVal + per2
       theFtab.SetValue(thePer2field,rec,per2)
     else
       per2 = (100 - cumPerVal)
       cumPerVal = cumPerVal + per2
       theFtab.SetValue(thePer2field,rec,per2)
     end
   else
     if (cumPerVal > 0) then  'if continuing
      if (theDepthVal < theLay3Val) then
        per2 =  (((theLay3Val - theLay2Val) / screenLengthVal) * 100).abs
        cumPerVal = cumPerVal + per2
        theFtab.SetValue(thePer2field,rec,per2)
      else
        per2 =  (((theDepthVal - theLay2Val) / screenLengthVal) * 100).abs
        cumPerVal = cumPerVal + per2
        theFtab.SetValue(thePer2field,rec,per2)
      end
     else
      per2 = 0
      theFtab.SetValue(thePer2field,rec,per2)
     end
   end
  end
'---------------------------------------------layer 3
 if (cumperval.round = 100) then
     'continue
     ct=ct+1
     per3 = 0
     theFtab.SetValue(thePer3field,rec,per3)
  else
   if ((theScreenVal < theLay3Val ) And (theScreenVal > theLay4Val)) then
     if (theDepthVal < theLay4Val) then
       per3 =  (((theScreenVal - theLay4Val) / screenLengthVal) * 100).abs
       cumPerVal = cumPerVal + per3
       theFtab.SetValue(thePer3field,rec,per3)
     else
       per3 = (100 - cumPerVal)
       cumPerVal = cumPerVal + per3
       theFtab.SetValue(thePer3field,rec,per3)
```

```
      end
    else
      if (cumPerVal > 0) then  'if continuing
        if (theDepthVal < theLay4Val) then
          per3 =  (((theLay4Val - theLay3Val) / screenLengthVal) * 100).abs
          cumPerVal = cumPerVal + per3
          theFtab.SetValue(thePer3field,rec,per3)
        else
          per3 =  (((theDepthVal - theLay3Val) / screenLengthVal) * 100).abs
          cumPerVal = cumPerVal + per3
          theFtab.SetValue(thePer3field,rec,per3)
        end
      else
        per3 = 0
        theFtab.SetValue(thePer3field,rec,per3)
      end
    end
  end
'---------------------------------------------layer 4
  if (cumperval.round = 100) then
     'continue
     ct=ct+1
     per4 = 0
     theFtab.SetValue(thePer4field,rec,per4)
  else
    if ((theScreenVal < theLay4Val ) And (theScreenVal > theLay5Val)) then
      if (theDepthVal < theLay5Val) then
        per4 =  (((theScreenVal - theLay5Val) / screenLengthVal) * 100).abs
        cumPerVal = cumPerVal + per4
        theFtab.SetValue(thePer4field,rec,per4)
      else
        per4 = (100 - cumPerVal)
        cumPerVal = cumPerVal + per4
        theFtab.SetValue(thePer4field,rec,per4)
      end
    else
      if (cumPerVal > 0) then  'if continuing
        if (theDepthVal < theLay5Val) then
          per4 =  (((theLay5Val - theLay4Val) / screenLengthVal) * 100).abs
          cumPerVal = cumPerVal + per4
          theFtab.SetValue(thePer4field,rec,per4)
        else
          per4 =  (((theDepthVal - theLay4Val) / screenLengthVal) * 100).abs
          cumPerVal = cumPerVal + per4
          theFtab.SetValue(thePer4field,rec,per4)
        end
      else
        per4 = 0
        theFtab.SetValue(thePer4field,rec,per4)
      end
    end
  end
'---------------------------------------------layer 5
  if (cumperval.round = 100) then
     'continue
     ct = ct+1
```

```
      per5 = 0
      theFtab.SetValue(thePer5field,rec,per5)
    else
     if ((theScreenVal < theLay5Val ) And (theScreenVal > theLay6Val)) then
       if (theDepthVal < theLay6Val) then
         per5 =  (((theScreenVal - theLay6Val) / screenLengthVal) * 100).abs
         cumPerVal = cumPerVal + per5
         theFtab.SetValue(thePer5field,rec,per5)
       else
         per5 = (100 - cumPerVal)
         cumPerVal = cumPerVal + per5
         theFtab.SetValue(thePer5field,rec,per5)
       end
     else
       if (cumPerVal > 0) then  'if continuing
        if (theDepthVal < theLay6Val) then
          per5 =  (((theLay6Val - theLay5Val) / screenLengthVal) * 100).abs
          cumPerVal = cumPerVal + per5
          theFtab.SetValue(thePer5field,rec,per5)
        else
          per5 =  (((theDepthVal - theLay5Val) / screenLengthVal) * 100).abs
          cumPerVal = cumPerVal + per5
          theFtab.SetValue(thePer5field,rec,per5)
        end
       else
        per5 = 0
        theFtab.SetValue(thePer5field,rec,per5)
       end
     end
    end
'-------------------------------------------layer 6
 if (cumPerVal.round = 100) then
      'continue
      ct = ct+1
      per6 = 0
      theFtab.SetValue(thePer6field,rec,per6)
    else
     if ((theScreenVal < theLay6Val ) And (theScreenVal > theBotVal)) then
       if (theDepthVal < theBotVal) then
         per6 =  (((theScreenVal - theBotVal) / screenLengthVal) * 100).abs
         cumPerVal = cumPerVal + per6
         theFtab.SetValue(thePer6field,rec,per6)
       else
         per6 = (100 - cumPerVal)
         cumPerVal = cumPerVal + per6
         theFtab.SetValue(thePer6field,rec,per6)
       end
     else
       if (cumPerVal > 0) then  'if continuing
        if (theDepthVal < theBotVal) then
          per6 =  (((theBotVal - theLay6Val) / screenLengthVal) * 100).abs
          cumPerVal = cumPerVal + per6
          theFtab.SetValue(thePer6field,rec,per6)
        else
          per6 =  (((theDepthVal - theLay6Val) / screenLengthVal) * 100).abs
          cumPerVal = cumPerVal + per6
```

```
        theFtab.SetValue(thePer6field,rec,per6)
      end
     else
      per6 = 0
      theFtab.SetValue(thePer6field,rec,per6)
     end
   end
  end
theFtab.SetValue(theTotPerField,rec,cumPerVal)
end 'end for loop
```

Appendix 2 – Arcview script to return land surface elevation for a well from a DEM grid

```
'-----------------------------------------------------
' Name: getgridvalue.ave
' Date: 991004
'
' Description: Moves copies values from a grid to a
' feature theme. The values from the grid are placed
' in a user defined field. If the feature theme isn't
' a point theme, then the feature gets the grid value
' from the value under it's centroid point.
'
' Requires: Spatial Analyst
'
'
' Author: Originally written by Mikael Elmquist (mikael@swegis.com), but later
' modified by Jeremy Davies (jeremy.davies@noaa.gov)
'
'-----------------------------------------------------


theView = av.GetActiveDoc
theThemes={}



'-----------------
'Choose in theme
'-----------------
themeList = theView.GetThemes
rep = 0
stupid = 0
while (rep = 0)
 theTheme = MsgBox.ChoiceAsString(themeList,"Select theme that shall get values from the grid
theme.","GetGridValue")
 if (theTheme = NIL) then
  exit
 end
 if (theTheme.Is(Ftheme).Not) then
  stupid = stupid+1
  if (stupid = 4) then
   msgBox.Info("Dear ArcView GIS user. Try to select a valid theme","Problem?")
  end
  msgBox.Error("Not a valid theme","Error")
 else
  rep = 1
  theFtab = theTheme.GetFtab
 end
end
rep = 0
stupid = 0


theThemes={}
if (theFtab.CanEdit) then
 theFTab.SetEditable(true)
```

```
  if ((theFTab.CanAddFields).Not) then
    MsgBox.Info("Can't add fields to the table."+NL+"Check write permission.","Can't add grid values")
    exit
  end
else
  MsgBox.Info("Can't modify the feature table."+NL+
  "Check write permission.","Can't add grid values")
  exit
end

'-----------------
'Choose grid theme
'-----------------

for each TargetTheme in theView.GetThemes
  if (TargetTheme.Is(Gtheme)) then
    theThemes.Add(TargetTheme)
  end
end
theGtheme = MsgBox.ChoiceAsString(theThemes,"Select grid that shall assign values to the point
theme.","GetGridValue")
if (theGtheme = Nil) then
  exit
end
theGrid = theGtheme.Clone.GetGrid.Clone
thePrj = Prj.MakeNull


'------------------
' Add the new field
'------------------

'enter name of new field name and parameters
newField = MsgBox.Input( "Enter new field name:", "Value", "" )
fieldsize = MsgBox.Input( "Enter new field width:", "Value", "10" )
precision = MsgBox.Input( "Enter number of decimals places in new field:", "Value", "4" )

gridvalueField = Field.Make (newField,#FIELD_DECIMAL,fieldsize.asNumber,precision.asNumber)
theShapeField = theFtab.FindField("shape")
theFTab.AddFields({gridvalueField})


'------------------
' Copy values
'------------------
av.ShowMsg("Calculating values")
av.SetStatus(0)
sstatus = theFtab.GetNumRecords.Clone
for each aRec in theFtab
  av.SetStatus(aRec/sstatus*100)
  theValue = theGrid.CellValue(theFtab.returnValue(theShapeField,aRec).ReturnCenter,thePrj)
  av.SetStatus(aRec/sstatus*100)
  if (theValue<>Nil) then
    theFtab.SetValue(gridvalueField,aRec,theValue)
  end
```

```
end


'------------------
'Reset arcview
'------------------
theFtab.Flush
theFtab.Refresh
theFTab.SetEditable(False)
av.purgeobjects
av.ClearStatus
av.ClearMsg
```

APPENDIX C

Standard Operating Procedures (SOPs) for Processing Predictive Pumpage Data TWDB Groundwater Availability Modeling (GAM) Projects

Standard Operating Procedures (SOPs) for Processing Predictive Pumpage Data TWDB Groundwater Availability Modeling (GAM) Projects

TABLE OF CONTENTS

1. Background – These procedures were developed to further implement the guidance provided by the Texas Water Development Board (TWDB) in their Technical Memorandum 02-01 "Development of Predictive Pumpage Data Set for GAM." The information in that technical memorandum will not be repeated here, and readers should first consult that document.

2. Groundwater Use Source Data - To the extent possible, procedures for predictive pumpage distribution among model grid cells mimiced the procedures for historical pumpage data. Predicted future groundwater use estimates are derived from one spreadsheet **(GAMPredictivePumpage_2002SWP.xls)** provided by the Texas Water Development Board (TWDB), as well as the previously developed historical pumpage datasets. This spreadsheet contains water use estimates from the state water plans for each water user group for the years 2000, 2010, 2020, 2030, 2040, and 2050. Water user groups are generally assigned for each water user category (IRR, STK, MIN, MFG, PWR, MUN, and C-O) in each county-basin. However, individual municipal water supplies within a county-basin are assigned identified as separate water user groups. The water use categories are listed below:

   - IRR – irrigation
   - STK – livestock
   - MIN - mineral extraction
   - MFG – manufacturing
   - PWR – power generation
   - MUN – municipal water supply, and
   - C-O – county-other (rural domestic) use.

   Historical groundwater use records from the categories MIN, MFG, PWR, and MUN are available for each specific water user, each assigned an alphanumeric water user code (aka "alphanum") in historical water use data tables. Specific locations and wells from which this groundwater was pumped were identified in historical pumpage records. These are known as "well-specific" water use categories. However, the particular locations of historical groundwater pumpage were generally not known for the use categories IRR, STK, and C-O. These categories are known as "non-well-specific" water use categories. This pumpage was distributed spatially based on population density, land use, and other factors.

   The spreadsheet **GAMPredictivePumpage_2002SWP.xls** was downloaded from the TWDB web site. The spreadsheet file was then imported into a new Microsoft Access database file **Predictive Pumpage**.

3. Initial Processing

   3.1. Create a sub-set of data for the modeled aquifer and geographic area – The table

Predictive Pumpage_2002SWP was queried for water use in the aquifer of interest based on the aquifer's major aquifer code, as well as the code "99." Other records were deleted. Next, the table was queried for those records within source county ID's found in the modeling domain. Records for water pumpage outside the model domain were deleted.

3.2. Split water use between surface and ground water – Some records contain an aggregate of surface and ground water use, as indicated by a value of "04" in the field "SO_TYPE_ID_NEW." A new field "PERCENT GROUNDWATER" was added to the table and assigned a value from 0 to 1 based on information in the field "ADDTL COMMENTS."

3.3. Interpolate pumpage estimates for all years 2000-2050 – The table **Predictive Pumpage_2002SWP** only contains water use estimates for the years 2000, 2010, 2020, 2030, 2040, and 2050. Water use estimates for the intervening years are calculated by linear interpolation. This can be calculated in a query as for example:

$$Pumpage_{2001} = Pumpage_{2000} + modulus(2001,10)*[(Pumpage_{2010}-Pumpage_{2000})/10]$$

4. Spatially distribute well-specific pumpage –

4.1. Identify locations of new wells – If the field "Possible_New_Wells" contained a flag "NW", it was necessary to identify the location of the new wells. The Regional Water Plan was consulted to identify the location of the new wells (a map showing the projected locations of the new wells was available). Using Arcview, the latitude and longitude of the well(s) were estimated and copied into a new field "KD_comment." This latitude and longitude were used to identify the model grid_id(s) from which the well was expected to pump. These grid_id's were copied into a new field "grid_id" in the predictive pumpage table.

4.2. Matching Predictive to Historical Locations by "Alphanum" - We assumed that a water user would tend to pump water in the future from the same locations from which they had pumped groundwater historically. A specific water user can best be identified in the TWDB predictive pumpage data using the field "WUG_Prime_Alpha", or, if the water was purchased, the field "Seller Alpha."

4.2.1. A new field "Source_Alpha" was created and populated with the value from the field "WUG_Prime_Alpha" or, if available, the value from the field "Seller Alpha."

4.2.2. In many cases, no value of alpha_num was provided in the table for a well-specific WUG_ID, typically for MIN, MFG, and PWR. Therefore, the value(s) of "alphanum" associated with that WUG_ID in the historical pumpage table was copied to the predictive pumpage table.

In the case that multiple values of "alphanum" were identified for a given "WUG_ID" in the historical data, we first made replicate copies of the record in the predictive pumpage table for each value of alphanum, copied each alphanum into the field "Source_Alpha", and entered in the field "percent groundwater" the fraction of pumpage for each alphanum for the period 1995-1999 from the historical table. An explanation was entered in the field "KD_comment."

4.2.3. The value of "Source_Alpha" was matched manually to the field "alphanum" in the historical pumpage datasets, and the model grid_id identified for this water user in historical pumpage distribution was manually copied to the field "Grid_ID" in the predictive pumpage table.

In many cases, more than one grid was associated with a given "alphanum". The predictive pumpage for each alphanum was distributed among multiple Grid ID's in an identical manner as the average for the period 1995-1999. Additional copies of predictive pumpage records were added to equal the number of grid_id's, and a field "grid_frac" was added to the predictive pumpage table, and assigned a value from 0 to 1, calculated as the average of the 1995-1999 fraction of pumpage from that grid_id for that alphanum in the historical pumpage dataset. The values of grid_frac summed to 1 for each "source_alpha."

4.3. Create new tables for each well-specific water use category –

4.3.1. Create a new table or query for the water use category MUN containing a value of MUN pumpage for each grid_id for each year from 2000 to 2050. The pumpage for each record is calculated as the total pumpage for the year of interest multiplied by the fields "grid_frac" and "percent groundwater."

4.3.2. Create a new table or query for the water use category MFG containing a value of MFG pumpage for each grid_id for each year from 2000 to 2050. The pumpage for each record is calculated as the total pumpage for the year of interest multiplied by the fields "grid_frac" and "percent groundwater."

4.3.3. Create a new table or query for the water use category MIN containing a value of MIN pumpage for each grid_id for each year from 2000 to 2050. The pumpage for each record is calculated as the total pumpage for the year of interest multiplied by the fields "grid_frac" and "percent groundwater."

4.3.4. Create a new table or query for the water use category PWR containing a value of PWR pumpage for each grid_id for each year from 2000 to 2050. The pumpage for each record is calculated as the total pumpage for the year of interest multiplied by the fields "grid_frac" and "percent groundwater."

5. Spatially distribute non-well-specific pumpage – We assume that groundwater pumpage in the future would be distributed within each county-basin in a similar way that it has been done in the recent past. While we do not discount the impact of changes in population and land use due to urban growth, sprawl, and other factors, we cannot reliably predict the spatial locations of these changes.

5.1. Calculate the fraction of groundwater pumpage for "C-O" use from each grid cell within a county-basin from 1999.

5.1.1. Run a query to summarize "C-O" groundwater pumpage in 1999 for each county-basin within the model domain.

5.1.2. For each grid_id within each county-basin, divide the "C-O" pumpage value for the year 1999 by the total "C-O" pumpage for that county-basin. Save this as a new field "Fr_pumpage" for each grid_id.

5.1.3. As a quality check, sum the values of "Fr_pumpage" for C-O by county-basin to ensure they sum to 1.

5.1.4. Create a new table or query for the water use category "C-O" containing a value of C-O pumpage for each grid_id for each year from 2000 to 2050. The pumpage for each record is calculated as the total pumpage for the year of interest (from the TWDB-provided table **GAMPredictivePumpage_2002SWP.xls**, with interpolated values for intervening years) multiplied by the fields "percent groundwater" (from the same table) and the field "Fr_pumpage" from the previous three steps.

5.2. Calculate the fraction of groundwater pumpage for "IRR" use from each grid cell within a county-basin from 1999.

5.2.1. Run a query to summarize "IRR" groundwater pumpage in 1999 for each county-basin within the model domain.

5.2.2. For each grid_id within each county-basin, divide the "IRR" pumpage value for the year 1999 by the total "IRR" pumpage for that county-basin. Save this as a new field "Fr_pumpage" for each grid_id.

5.2.3. As a quality check, sum the values of "Fr_pumpage" for IRR by county-basin to ensure they sum to 1.

5.2.4. Create a new table or query for the water use category "IRR" containing a value of IRR pumpage for each grid_id for each year from 2000 to 2050. The pumpage for each record is calculated as the total pumpage for the year of interest (from the TWDB-provided table **GAMPredictivePumpage_2002SWP.xls**, with interpolated values for intervening years) multiplied by the fields "percent groundwater" (from the same table) and the field "Fr_pumpage" from the previous three steps.

5.3. Calculate the fraction of groundwater pumpage for "STK" use from each grid cell within a county-basin from 1999.

5.3.1. Run a query to summarize "STK" groundwater pumpage in 1999 for each county-basin within the model domain.

5.3.2. For each grid_id within each county-basin, divide the "STK" pumpage value for the year 1999 by the total "STK" pumpage for that county-basin. Save this as a new field "Fr_pumpage" for each grid_id.

5.3.3. As a quality check, sum the values of "Fr_pumpage" for STK by county-basin to ensure they sum to 1.

5.3.4. Create a new table or query for the water use category "STK" containing a value of STK pumpage for each grid_id for each year from 2000 to 2050. The pumpage for each record is calculated as the total pumpage for the year of interest (from the TWDB-provided table **GAMPredictivePumpage_2002SWP.xls**, with interpolated values for intervening years) multiplied by the fields "percent groundwater" (from the same table) and the field "Fr_pumpage" from the previous three steps.

5.4. Note: The result of this step should be three tables (or queries), one each for C-O, IRR, and STK. Each should contain, at a minimum, the fields "Grid_ID", "county_name", "basin_name", "year", "data_cat", and "pumpage."

6. Monthly Distribution of Annual Pumpage Totals - We assume that the historical average of monthly water use distribution is a valid predictor of future monthly distribution.

   Monthly factors are calculated for each county-basin and data_cat as the average of mfraction for the period 1995-1999 (in the historical pumpage table "MONTHLY FACTORS") in a new table **PredictiveMonthlyFactors**. There should be a monthly factor for each combination of the seven water use categories and county-basin. If no monthly factor can be calculated because there was no historical pumpage, then the monthly factor for that data_cat in the nearest other county-basin should be used.

7. Summarize Pumpage Information to Create Model Input Files - Summary queries for a given year and/or month should be performed as described in the SOP for historical pumpage data.

8. Handling Non-Texas Pumpage – Predictions of future pumpage for portions of the model domain outside of Texas are not available from the Texas Regional Water Plans. In this case, we will assume that the average pumpage for the period 1995-1999 is the best estimate of future pumpage for the water use categories MFG, MIN, PWR, STK, and IRR. Because population projections are available, however, we can project future water use for MUN and C-O based on the 1990 water use for each county or parish and the ratio of projected future county/parish population to its 1990 population.

   8.1. Download from the respective state census data center or the U.S. census bureau population estimates from each county or parish through 2050. Linearly interpolate values for intervening years if necessary.

   8.2. For each year from 2000 to 2050, calculate the ratio of projected population for each year to that in 2000 for each county or parish.

   8.3. Multiply the historical pumpage value from C-O or MUN out-of-Texas records in 1999 by the factor to obtain a projected pumpage estimate for that year.

APPENDIX D1

Tabulated Groundwater Withdrawal Estimates for the Carrizo-Wilcox for 1980, 1990, 2000, 2010, 2020, 2030, 2040, and 2050

Table D1.1
Rate of groundwater withdrawal (acre-feet per year) from flow layer 1 of the Carrizo-Wilcox aquifer for counties within the study area

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	1,198	1,876	2,663	2,723	2,753	2,789	2,781	2,842
Angelina	16,322	13,746	2,309	2,144	2,151	2,257	2,287	2,419
Bienville, LA	-	-	669	669	669	669	669	669
Bossier, LA	-	-	1,634	1,728	1,817	1,901	1,981	2,056
Bowie	-	-	-	-	-	-	-	-
Caddo, LA	9	545	7	7	7	7	8	8
Camp	31	42	60	70	71	71	71	72
Cass	1,124	1,305	291	302	309	317	355	361
Cherokee	1,255	1,732	2,484	908	973	1,027	1,077	1,121
De Soto, LA	-	-	-	-	-	-	-	-
Franklin	-	-	-	-	-	-	-	-
Freestone	24	21	22	21	20	20	20	20
Gregg	382	356	173	238	237	249	259	267
Grimes	383	733	411	437	465	498	475	538
Harrison	242	370	304	304	344	354	351	350
Henderson	533	911	661	673	673	667	655	672
Hopkins	-	-	-	-	-	-	-	-
Houston	1,581	1,726	703	767	765	766	770	772
Leon	451	683	1,070	1,094	1,121	1,150	1,181	1,217
Limestone	-	-	-	-	-	-	-	-
Madison	815	944	326	324	318	313	305	297
Marion	398	477	339	363	390	419	450	486
Miller, AR	-	363	10	10	10	10	10	10
Morris	315	325	39	39	37	37	36	34
Nacogdoches	1,658	1,620	785	793	805	516	813	818
Natchitoches, LA	613	458	929	948	975	1,008	1,050	1,098
Navarro	-	-	-	-	-	-	-	-
Panola	-	-	-	-	-	-	-	-
Rains	-	-	-	-	-	-	-	-
Red River, LA	-	-	699	724	777	858	968	1,107
Robertson	-	2	24	23	21	21	21	21
Rusk	55	65	68	69	74	80	80	85
Sabine, LA	961	1,141	1,842	1,977	2,122	2,281	2,452	2,635
Sabine, TX	792	1,045	1,025	1,094	1,158	1,272	1,340	1,369
San Augustine	6,089	4,468	253	250	246	249	247	249
Shelby	-	-	-	-	-	-	-	-
Smith	4,502	6,605	8,431	8,643	9,470	121	122	122
Titus	23	23	19	25	26	29	30	30
Trinity	1,819	1,816	-	-	-	-	-	-
Upshur	1,144	1,547	1,315	1,401	1,402	1,413	1,378	1,381
Van Zandt	176	224	214	221	250	260	262	262
Wood	756	1,163	1,345	1,476	1,539	1,645	1,722	1,890
Total	43,651	46,332	31,124	30,465	31,995	23,274	24,226	25,278

Table D1.2
Rate of groundwater withdrawal (acre-feet per year) from flow layer 2 of the Carrizo-Wilcox aquifer for counties within the study area

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	317	431	611	623	629	636	634	646
Angelina	8	9	4	4	4	4	4	4
Bienville, LA	-	-	-	-	-	-	-	-
Bossier, LA	1	1	1	1	1	1	1	1
Bowie	-	-	-	-	-	-	-	-
Caddo, LA	11	7	6	7	7	7	8	9
Camp	23	28	51	56	56	57	57	57
Cass	375	488	407	111	113	116	127	129
Cherokee	658	955	1,364	550	584	611	637	599
De Soto, LA	-	-	-	-	-	-	-	-
Franklin	1	4	-	-	-	-	-	-
Freestone	35	37	44	43	41	40	40	40
Gregg	224	202	69	113	112	116	119	121
Grimes	-	-	-	-	-	-	-	-
Harrison	442	605	424	425	474	484	487	485
Henderson	155	223	230	234	236	236	233	236
Hopkins	-	-	-	-	-	-	-	-
Houston	34	37	22	23	22	22	23	23
Leon	167	259	520	526	534	542	550	560
Limestone	-	-	-	-	-	-	-	-
Madison	3	3	205	205	205	205	205	205
Marion	70	107	63	65	66	69	68	69
Miller, AR	-	23	44	44	44	44	44	44
Morris	295	312	38	38	36	36	34	33
Nacogdoches	816	1,017	1,089	1,098	1,101	1,020	1,113	1,126
Natchitoches, LA	54	68	91	94	98	103	108	115
Navarro	-	-	-	-	-	-	-	-
Panola	-	-	-	-	-	-	-	-
Rains	-	-	-	-	-	-	-	-
Red River, LA	-	-	-	-	-	-	-	-
Robertson	-	-	-	-	-	-	-	-
Rusk	97	119	124	127	135	145	146	154
Sabine, LA	-	-	-	-	-	-	-	-
Sabine, TX	-	-	-	-	-	-	-	-
San Augustine	78	68	40	40	39	39	39	39
Shelby	-	-	-	-	-	-	-	-
Smith	457	633	829	847	928	1,027	1,137	1,250
Titus	37	54	38	54	57	63	67	69
Trinity	-	-	-	-	-	-	-	-
Upshur	690	687	365	393	394	403	406	408
Van Zandt	24	34	31	32	36	37	37	37
Wood	244	310	320	350	365	389	407	445
Total	5,316	6,721	7,030	6,103	6,317	6,452	6,731	6,904

Table D1.3
Rate of groundwater withdrawal (acre-feet per year) from flow layer 3 of the Carrizo-Wilcox aquifer for counties within the study area

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	713	1,021	1,098	1,110	1,112	1,119	1,114	1,127
Angelina	5,592	5,786	3,257	2,185	2,224	2,553	2,711	3,047
Bienville, LA	-	-	-	-	-	-	-	-
Bossier, LA	-	-	-	-	-	-	-	-
Bowie	-	-	-	-	-	-	-	-
Caddo, LA	92	40	25	25	26	27	28	30
Camp	191	230	309	380	385	391	394	396
Cass	210	271	58	59	60	61	65	66
Cherokee	2,144	2,027	2,221	828	868	974	1,018	1,105
De Soto, LA	-	-	-	-	-	-	-	-
Franklin	26	29	7	7	7	8	9	5
Freestone	43	31	54	54	54	54	54	54
Gregg	10	8	56	56	56	56	56	56
Grimes	-	-	331	340	351	366	394	429
Harrison	540	726	462	486	527	546	576	592
Henderson	332	520	460	469	472	370	362	472
Hopkins	-	-	-	-	-	-	-	-
Houston	295	16	704	665	670	676	680	682
Leon	395	606	2,170	1,827	1,342	1,246	1,222	1,253
Limestone	-	-	-	-	-	-	-	-
Madison	68	154	1,202	1,158	1,125	1,091	1,041	998
Marion	115	117	71	73	74	76	77	77
Miller, AR	-	525	931	931	931	931	931	931
Morris	304	103	9	9	9	9	8	8
Nacogdoches	5,087	5,669	2,751	2,601	2,734	2,929	3,135	3,421
Natchitoches, LA	28	30	35	35	36	36	36	37
Navarro	-	-	-	-	-	-	-	-
Panola	-	-	-	-	-	-	-	-
Rains	-	-	-	-	-	-	-	-
Red River, LA	-	-	-	-	-	-	-	-
Robertson	11	13	255	248	244	239	235	231
Rusk	495	591	473	441	472	511	515	544
Sabine, LA	-	-	-	-	-	-	-	-
Sabine, TX	-	-	-	-	-	-	-	-
San Augustine	176	193	123	123	123	124	125	126
Shelby	1	1	-	-	-	-	-	-
Smith	1,178	1,267	1,533	1,578	1,716	1,882	2,064	2,257
Titus	71	88	69	91	94	104	109	111
Trinity	-	-	-	-	-	-	-	-
Upshur	854	760	598	630	628	644	653	667
Van Zandt	82	144	130	136	168	166	168	168
Wood	1,267	1,049	718	764	810	867	909	988
Total	20,320	22,015	20,110	17,309	17,318	18,056	18,689	19,878

Table D1.4
Rate of groundwater withdrawal (acre-feet per year) from flow layer 4 of the Carrizo-Wilcox aquifer for counties within the study area

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	992	1,269	2,173	2,127	2,065	2,052	2,021	2,048
Angelina	601	649	12,237	11,841	10,698	11,298	11,992	13,208
Bienville, LA	-	-	-	-	-	-	-	-
Bossier, LA	11	5	8	8	8	8	8	8
Bowie	1	3	-	-	-	-	-	-
Caddo, LA	592	195	464	473	490	517	554	600
Camp	675	879	814	1,025	1,041	1,059	1,070	1,077
Cass	459	513	276	571	266	263	261	258
Cherokee	3,033	3,039	2,536	2,007	1,990	1,940	2,078	2,215
De Soto, LA	-	-	-	-	-	-	-	-
Franklin	119	160	75	76	76	77	80	80
Freestone	410	448	510	495	474	467	470	469
Gregg	1,770	1,515	1,072	1,213	1,216	1,272	1,320	1,363
Grimes	-	-	-	-	-	-	-	-
Harrison	586	744	559	593	640	655	674	683
Henderson	711	1,106	934	942	938	935	929	944
Hopkins	45	49	33	42	41	45	50	54
Houston	2	2	11	11	11	11	11	11
Leon	1,021	1,440	2,145	2,172	2,200	2,296	2,386	2,510
Limestone	18	33	117	118	120	125	130	136
Madison	4	10	-	-	-	-	-	-
Marion	257	246	153	156	160	162	164	165
Miller, AR	-	1,122	1,945	1,945	1,945	1,945	1,945	1,945
Morris	452	6,681	26	26	25	25	24	23
Nacogdoches	779	958	2,122	2,020	2,097	2,229	2,367	2,563
Natchitoches, LA	289	350	574	592	619	652	692	739
Navarro	-	-	-	-	-	-	-	-
Panola	19	20	11	11	11	10	10	10
Rains	-	-	-	-	-	-	-	-
Red River, LA	-	-	-	-	-	-	-	-
Robertson	198	200	7,777	7,615	7,550	7,372	7,206	7,047
Rusk	3,276	3,884	3,849	3,270	3,347	3,540	3,589	3,733
Sabine, LA	-	-	-	-	-	-	-	-
Sabine, TX	-	-	-	-	-	-	-	-
San Augustine	266	267	141	142	142	145	145	146
Shelby	1,220	1,358	1,289	1,482	1,238	1,001	1,582	1,819
Smith	4,338	2,868	5,130	5,642	6,031	6,075	6,544	5,222
Titus	272	363	309	396	410	448	466	477
Trinity	-	-	-	-	-	-	-	-
Upshur	807	939	843	890	894	913	605	965
Van Zandt	504	715	1,021	1,099	1,284	1,318	1,376	1,445
Wood	1,465	1,007	1,507	1,618	1,761	1,911	2,061	2,273
Total	25,192	33,037	50,661	50,618	49,788	50,766	52,810	54,236

Table D1.5
Rate of groundwater withdrawal (acre-feet per year) from flow layer 5 of the Carrizo-Wilcox aquifer for counties within the study area

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	209	20	179	188	196	202	215	226
Angelina	-	-	-	-	-	-	-	-
Bienville, LA	-	-	-	-	-	-	-	-
Bossier, LA	107	67	80	83	86	88	90	92
Bowie	1,082	1,274	643	681	681	681	681	681
Caddo, LA	3,389	2,405	2,858	2,932	3,084	3,313	3,619	4,003
Camp	477	527	301	299	302	307	314	322
Cass	1,090	1,064	57	56	55	53	52	52
Cherokee	3	37	108	28	30	32	34	37
De Soto, LA	1,290	1,053	226	226	226	226	226	226
Franklin	654	714	1,675	1,581	1,535	1,476	1,500	1,562
Freestone	382	1,143	509	523	507	497	499	499
Gregg	431	282	465	448	433	435	439	442
Grimes	-	-	-	-	-	-	-	-
Harrison	1,433	1,729	1,341	1,358	1,481	1,518	1,559	1,584
Henderson	1,031	1,105	883	863	859	867	877	899
Hopkins	657	959	790	910	909	941	1,003	1,037
Houston	-	-	-	-	-	-	-	-
Leon	-	-	-	-	-	-	-	-
Limestone	107	805	6,919	7,611	7,622	7,643	7,666	7,694
Madison	-	-	-	-	-	-	-	-
Marion	80	96	151	125	113	108	105	119
Miller, AR	24	6,736	4,234	4,236	4,238	4,238	4,240	4,242
Morris	575	395	590	593	582	576	564	560
Nacogdoches	358	360	391	395	395	420	435	453
Natchitoches, LA	137	112	155	155	156	157	157	159
Navarro	10	16	-	-	-	-	-	-
Panola	3,067	4,172	2,005	1,960	1,890	1,962	1,963	1,960
Rains	297	465	368	389	408	276	293	311
Red River, LA	23	76	174	174	175	176	177	179
Robertson	173	50	6,450	6,295	6,212	6,055	5,917	5,781
Rusk	3,241	3,249	3,464	3,123	2,798	2,601	2,512	2,464
Sabine, LA	-	-	-	-	-	-	-	-
Sabine, TX	-	-	-	-	-	-	-	-
San Augustine	-	-	-	-	-	-	-	-
Shelby	1,740	1,794	1,984	2,262	1,852	2,102	2,386	2,750
Smith	1,073	653	2,261	2,486	2,655	2,669	2,839	2,243
Titus	989	1,229	2,698	2,743	2,731	2,784	2,817	2,846
Trinity	-	-	-	-	-	-	-	-
Upshur	85	110	106	110	109	110	110	110
Van Zandt	2,132	1,919	1,477	1,622	2,219	2,049	2,119	2,195
Wood	368	622	704	767	797	848	886	967
Total	26,714	35,238	44,246	45,222	45,336	45,410	46,294	46,695

Table D1.6
Rate of groundwater withdrawal (acre-feet per year) from flow layer 6 of the Carrizo-Wilcox aquifer for counties within the study area

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	64	84	16	17	17	18	18	19
Angelina	-	-	-	-	-	-	-	-
Bienville, LA	-	-	-	-	-	-	-	-
Bossier, LA	9	2	5	5	5	5	5	5
Bowie	841	914	224	1,264	1,265	1,267	1,271	1,276
Caddo, LA	930	614	619	634	664	711	772	849
Camp	-	5	7	7	7	7	7	7
Cass	645	656	202	340	335	330	325	309
Cherokee	-	-	-	-	-	-	-	-
De Soto, LA	615	327	5	5	5	5	5	5
Franklin	307	428	275	276	276	276	278	278
Freestone	1,514	1,657	1,881	1,903	1,931	1,975	2,001	2,025
Gregg	-	-	356	372	387	409	432	459
Grimes	-	-	-	-	-	-	-	-
Harrison	406	318	398	506	557	591	599	620
Henderson	1,373	1,797	2,002	1,741	1,740	1,747	1,751	1,768
Hopkins	1,430	1,970	989	1,092	1,092	1,106	1,140	1,155
Houston	-	-	-	-	-	-	-	-
Leon	-	-	-	-	-	-	-	-
Limestone	243	339	1,441	1,448	1,472	1,516	1,564	1,623
Madison	-	-	-	-	-	-	-	-
Marion	2	-	-	-	-	-	-	-
Miller, AR	2	11	21	22	22	22	23	23
Morris	4	5	16	16	16	16	16	16
Nacogdoches	-	-	1	1	1	1	1	1
Natchitoches, LA	-	-	-	-	-	-	-	-
Navarro	57	99	12	12	12	12	12	12
Panola	401	446	1,861	1,608	1,360	2,180	2,205	2,178
Rains	90	153	-	-	-	-	-	-
Red River, LA	1	23	59	59	59	59	59	59
Robertson	-	-	-	-	-	-	-	-
Rusk	74	4	995	895	794	760	756	760
Sabine, LA	-	-	-	-	-	-	-	-
Sabine, TX	-	-	-	-	-	-	-	-
San Augustine	-	-	-	-	-	-	-	-
Shelby	21	29	156	152	149	149	150	154
Smith	-	-	-	-	-	-	-	-
Titus	108	138	60	60	60	61	61	61
Trinity	-	-	-	-	-	-	-	-
Upshur	-	-	-	-	-	-	-	-
Van Zandt	1,638	2,017	1,731	1,758	2,073	2,091	2,299	2,428
Wood	1	2	1,129	1,129	1,129	1,129	1,129	1,129
Total	10,776	12,038	14,461	15,322	15,428	16,443	16,879	17,219

Table D1.7

Rate of groundwater withdrawal (acre-feet per year) from flow layer 1 of the Carrizo-Wilcox Aquifer for counties within the study area

Municipal and Industrial*

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	-	-	-	-	-	-	-	-
Angelina	13,317	9,705	524	344	350	404	429	484
Bienville, LA	-	-	669	669	669	669	669	669
Bossier, LA	-	-	1,634	1,728	1,817	1,901	1,981	2,056
Bowie	-	-	-	-	-	-	-	-
Caddo, LA	-	540	-	-	-	-	-	-
Camp	-	-	-	-	-	-	-	-
Cass	-	-	-	-	-	-	-	-
Cherokee	-	-	-	-	-	-	-	-
De Soto, LA	-	-	-	-	-	-	-	-
Franklin	-	-	-	-	-	-	-	-
Freestone	-	-	-	-	-	-	-	-
Gregg	-	-	-	-	-	-	-	-
Grimes	-	-	-	-	-	-	-	-
Harrison	-	-	-	-	-	-	-	-
Henderson	5	6	-	-	-	-	-	-
Hopkins	-	-	-	-	-	-	-	-
Houston	-	-	2	3	3	4	5	5
Leon	-	-	-	-	-	-	-	-
Limestone	-	-	-	-	-	-	-	-
Madison	-	-	-	-	-	-	-	-
Marion	-	-	-	-	-	-	-	-
Miller, AR	-	360	-	-	-	-	-	-
Morris	-	-	-	-	-	-	-	-
Nacogdoches	1,145	815	-	-	-	-	-	-
Natchitoches, LA	-	-	285	301	324	352	387	428
Navarro	-	-	-	-	-	-	-	-
Panola	-	-	-	-	-	-	-	-
Rains	-	-	-	-	-	-	-	-
Red River, LA	-	-	699	724	777	858	968	1,107
Robertson	-	1	-	-	-	-	-	-
Rusk	-	-	-	-	-	-	-	-
Sabine, LA	-	569	979	1,055	1,137	1,227	1,324	1,428
Sabine, TX	-	-	786	843	895	946	996	1,045
San Augustine	5,723	4,075	-	-	-	-	-	-
Shelby	-	-	-	-	-	-	-	-
Smith	-	-	-	-	-	-	-	-
Titus	-	-	-	-	-	-	-	-
Trinity	1,145	815	-	-	-	-	-	-
Upshur	-	-	-	-	-	-	-	-
Van Zandt	-	-	-	-	-	-	-	-
Wood	-	-	-	-	-	-	-	-
Total	21,335	16,886	5,578	5,667	5,972	6,361	6,759	7,222

*industrial includes manufacturing, mining, and power generation

Table D1.7 (*Continued...*)

Rate of groundwater withdrawal (acre-feet per year) from flow layer 1 of the Carrizo-Wilcox Aquifer for counties within the study area

County – Other (Non-reported Domestic)

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	1,179	1,752	2,623	2,683	2,713	2,749	2,741	2,802
Angelina	3,005	4,041	1,785	1,800	1,801	1,853	1,858	1,935
Bienville, LA	-	-	-	-	-	-	-	-
Bossier, LA	-	-	-	-	-	-	-	-
Bowie	-	-	-	-	-	-	-	-
Caddo, LA	4	3	3	3	3	3	4	4
Camp	24	26	29	39	40	40	40	41
Cass	1,114	1,281	287	298	305	313	351	357
Cherokee	1,255	1,624	2,331	755	820	874	924	968
De Soto, LA	-	-	-	-	-	-	-	-
Franklin	-	-	-	-	-	-	-	-
Freestone	23	20	20	19	18	18	18	18
Gregg	382	356	173	238	237	249	259	267
Grimes	383	733	411	437	465	498	475	538
Harrison	242	370	303	303	343	353	350	349
Henderson	490	843	628	640	640	634	622	639
Hopkins	-	-	-	-	-	-	-	-
Houston	1,539	1,661	627	688	687	686	688	692
Leon	315	474	339	363	390	419	450	486
Limestone	-	-	-	-	-	-	-	-
Madison	810	939	170	168	162	157	149	141
Marion	398	477	339	363	390	419	450	486
Miller, AR	-	-	-	-	-	-	-	-
Morris	308	314	39	39	37	37	36	34
Nacogdoches	495	790	763	771	784	495	793	796
Natchitoches, LA	17	38	50	53	57	62	69	76
Navarro	-	-	-	-	-	-	-	-
Panola	-	-	-	-	-	-	-	-
Rains	-	-	-	-	-	-	-	-
Red River, LA	-	-	-	-	-	-	-	-
Robertson	-	1	24	23	21	21	21	21
Rusk	55	65	68	69	74	80	80	85
Sabine, LA	844	2	749	808	871	940	1,014	1,093
Sabine, TX	725	960	149	155	162	171	177	143
San Augustine	343	368	213	210	206	208	206	208
Shelby	-	-	-	-	-	-	-	-
Smith	4,463	6,560	8,332	8,544	9,371	22	23	23
Titus	20	19	12	18	19	22	23	23
Trinity	674	1,001	-	-	-	-	-	-
Upshur	1,137	1,535	1,289	1,375	1,376	1,387	1,352	1,355
Van Zandt	175	222	214	221	250	260	262	262
Wood	739	1,139	1,326	1,457	1,520	1,626	1,703	1,871
Total	21,158	27,614	23,296	22,540	23,762	14,596	15,138	15,713

Table D1.7 (*Continued...*)
Rate of groundwater withdrawal (acre-feet per year) from flow layer 1 of the Carrizo-Wilcox Aquifer for counties within the study area

Livestock

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	19	124	40	40	40	40	40	40
Angelina	-	-	-	-	-	-	-	-
Bienville, LA	-	-	-	-	-	-	-	-
Bossier, LA	-	-	-	-	-	-	-	-
Bowie	-	-	-	-	-	-	-	-
Caddo, LA	2	2	2	2	2	2	2	2
Camp	7	7	20	20	20	20	20	20
Cass	10	24	4	4	4	4	4	4
Cherokee	-	108	153	153	153	153	153	153
De Soto, LA	-	-	-	-	-	-	-	-
Franklin	-	-	-	-	-	-	-	-
Freestone	1	1	2	2	2	2	2	2
Gregg	-	-	-	-	-	-	-	-
Grimes	-	-	-	-	-	-	-	-
Harrison	-	-	-	-	-	-	-	-
Henderson	38	62	33	33	33	33	33	33
Hopkins	-	-	-	-	-	-	-	-
Houston	42	65	64	66	65	66	66	64
Leon	136	209	731	731	731	731	731	731
Limestone	-	-	-	-	-	-	-	-
Madison	2	2	156	156	156	156	156	156
Marion	-	-	-	-	-	-	-	-
Miller, AR	-	1	6	6	6	6	6	6
Morris	7	11	-	-	-	-	-	-
Nacogdoches	18	15	20	20	19	19	18	20
Natchitoches, LA	298	-	109	109	109	109	109	109
Navarro	-	-	-	-	-	-	-	-
Panola	-	-	-	-	-	-	-	-
Rains	-	-	-	-	-	-	-	-
Red River, LA	-	-	-	-	-	-	-	-
Robertson	-	-	-	-	-	-	-	-
Rusk	-	-	-	-	-	-	-	-
Sabine, LA	117	569	113	113	113	113	113	113
Sabine, TX	67	85	90	96	101	155	167	181
San Augustine	23	25	33	33	33	34	34	34
Shelby	-	-	-	-	-	-	-	-
Smith	39	45	98	98	98	98	98	98
Titus	3	4	7	7	7	7	7	7
Trinity	-	-	-	-	-	-	-	-
Upshur	7	12	26	26	26	26	26	26
Van Zandt	1	2	-	-	-	-	-	-
Wood	17	23	17	17	17	17	17	17
Total	854	1,396	1,724	1,732	1,735	1,791	1,802	1,816

Table D1.7 (*Continued...*)
Rate of groundwater withdrawal (acre-feet per year) from flow layer 1 of the Carrizo-Wilcox Aquifer for counties within the study area

Irrigation

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	-	-	-	-	-	-	-	-
Angelina	-	-	-	-	-	-	-	-
Bienville, LA	-	-	-	-	-	-	-	-
Bossier, LA	-	-	-	-	-	-	-	-
Bowie	-	-	-	-	-	-	-	-
Caddo, LA	3	-	2	2	2	2	2	2
Camp	-	9	11	11	11	11	11	11
Cass	-	-	-	-	-	-	-	-
Cherokee	-	-	-	-	-	-	-	-
De Soto, LA	-	-	-	-	-	-	-	-
Franklin	-	-	-	-	-	-	-	-
Freestone	-	-	-	-	-	-	-	-
Gregg	-	-	-	-	-	-	-	-
Grimes	-	-	-	-	-	-	-	-
Harrison	-	-	1	1	1	1	1	1
Henderson	-	-	-	-	-	-	-	-
Hopkins	-	-	-	-	-	-	-	-
Houston	-	-	10	10	10	10	11	11
Leon	-	-	-	-	-	-	-	-
Limestone	-	-	-	-	-	-	-	-
Madison	3	3	-	-	-	-	-	-
Marion	-	-	-	-	-	-	-	-
Miller, AR	-	2	4	4	4	4	4	4
Morris	-	-	-	-	-	-	-	-
Nacogdoches	-	-	2	2	2	2	2	2
Natchitoches, LA	298	420	485	485	485	485	485	485
Navarro	-	-	-	-	-	-	-	-
Panola	-	-	-	-	-	-	-	-
Rains	-	-	-	-	-	-	-	-
Red River, LA	-	-	-	-	-	-	-	-
Robertson	-	-	-	-	-	-	-	-
Rusk	-	-	-	-	-	-	-	-
Sabine, LA	-	1	1	1	1	1	1	1
Sabine, TX	-	-	-	-	-	-	-	-
San Augustine	-	-	7	7	7	7	7	7
Shelby	-	-	-	-	-	-	-	-
Smith	-	-	1	1	1	1	1	1
Titus	-	-	-	-	-	-	-	-
Trinity	-	-	-	-	-	-	-	-
Upshur	-	-	-	-	-	-	-	-
Van Zandt	-	-	-	-	-	-	-	-
Wood	-	1	2	2	2	2	2	2
Total	304	436	526	526	526	526	527	527

Table D1.8
Rate of groundwater withdrawal (acre-feet per year) from flow layer 2 of the Carrizo-Wilcox Aquifer for counties within the study area

Municipal and Industrial*

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	-	-	-	-	-	-	-	-
Angelina	-	-	-	-	-	-	-	-
Bienville, LA	-	-	-	-	-	-	-	-
Bossier, LA	-	-	-	-	-	-	-	-
Bowie	-	-	-	-	-	-	-	-
Caddo, LA	-	-	-	-	-	-	-	-
Camp	-	-	-	-	-	-	-	-
Cass	-	-	-	-	-	-	-	-
Cherokee	-	-	-	-	-	-	-	-
De Soto, LA	-	-	-	-	-	-	-	-
Franklin	-	-	-	-	-	-	-	-
Freestone	-	-	-	-	-	-	-	-
Gregg	-	-	-	-	-	-	-	-
Grimes	-	-	-	-	-	-	-	-
Harrison	-	-	-	-	-	-	-	-
Henderson	49	55	104	106	108	109	108	108
Hopkins	-	-	-	-	-	-	-	-
Houston	-	-	-	-	-	-	-	-
Leon	-	-	-	-	-	-	-	-
Limestone	-	-	-	-	-	-	-	-
Madison	-	-	-	-	-	-	-	-
Marion	-	-	-	-	-	-	-	-
Miller, AR	-	-	-	-	-	-	-	-
Morris	-	-	-	-	-	-	-	-
Nacogdoches	-	-	-	-	-	-	-	-
Natchitoches, LA	-	-	-	-	-	-	-	-
Navarro	-	-	-	-	-	-	-	-
Panola	-	-	-	-	-	-	-	-
Rains	-	-	-	-	-	-	-	-
Red River, LA	-	-	-	-	-	-	-	-
Robertson	-	-	-	-	-	-	-	-
Rusk	-	-	-	-	-	-	-	-
Sabine, LA	-	-	-	-	-	-	-	-
Sabine, TX	-	-	-	-	-	-	-	-
San Augustine	-	-	-	-	-	-	-	-
Shelby	-	-	-	-	-	-	-	-
Smith	-	-	-	-	-	-	-	-
Titus	-	-	-	-	-	-	-	-
Trinity	-	-	-	-	-	-	-	-
Upshur	312	166	-	-	-	-	-	-
Van Zandt	-	-	-	-	-	-	-	-
Wood	-	-	-	-	-	-	-	-
Total	361	221	104	106	108	109	108	108

*industrial includes manufacturing, mining, and power generation

Table D1.8 (*Continued...*)
Rate of groundwater withdrawal (acre-feet per year) from flow layer 2 of the Carrizo-Wilcox Aquifer for counties within the study area

County – Other (Non-reported Domestic)

County	1980	1990	2000	2010	2020	2030	2040	2050
Anderson	270	338	507	519	525	532	530	542
Angelina	8	9	4	4	4	4	4	4
Bienville, LA	-	-	-	-	-	-	-	-
Bossier, LA	-	-	-	-	-	-	-	-
Bowie	-	-	-	-	-	-	-	-
Caddo, LA	9	5	4	5	5	5	6	7
Camp	11	13	15	20	20	21	21	21
Cass	362	456	402	106	108	111	122	124
Cherokee	653	831	1,192	386	420	447	473	435
De Soto, LA	-	-	-	-	-	-	-	-
Franklin	1	4	-	-	-	-	-	-
Freestone	23	28	29	28	26	25	25	25
Gregg	224	202	69	113	112	116	119	121
Grimes	-	-	-	-	-	-	-	-
Harrison	433	602	423	423	472	482	485	483
Henderson	82	130	98	100	100	99	97	100
Hopkins	-	-	-	-	-	-	-	-
Houston	32	34	13	14	14	14	14	14
Leon	81	129	88	94	102	110	118	128
Limestone	-	-	-	-	-	-	-	-
Madison	-	-	-	-	-	-	-	-
Marion	70	107	63	65	66	69	68	69
Miller, AR	-	-	-	-	-	-	-	-
Morris	290	305	38	38	36	36	34	33
Nacogdoches	677	895	864	873	888	801	898	902
Natchitoches, LA	14	36	48	51	55	60	65	72
Navarro	-	-	-	-	-	-	-	-
Panola	-	-	-	-	-	-	-	-
Rains	-	-	-	-	-	-	-	-
Red River, LA	-	-	-	-	-	-	-	-
Robertson	-	-	-	-	-	-	-	-
Rusk	86	110	115	117	125	135	136	144
Sabine, LA	-	-	-	-	-	-	-	-
Sabine, TX	-	-	-	-	-	-	-	-
San Augustine	75	65	35	35	34	34	34	34
Shelby	-	-	-	-	-	-	-	-
Smith	444	619	792	810	891	990	1,100	1,213
Titus	35	51	33	49	52	58	62	64
Trinity	-	-	-	-	-	-	-	-
Upshur	376	517	356	384	385	394	397	399
Van Zandt	22	32	30	31	35	36	36	36
Wood	230	290	304	334	349	373	391	429
Total	4,508	5,808	5,522	4,599	4,824	4,952	5,235	5,399