Level 0 data

SNOTEL – flow through WOF webservice was written to return SNOTEL Data

Metadatada statically harvested except for sites. Sites has 2 harvesters. One goes to <http://www.wcc.nrcs.usda.gov/nwcc/sitelist.jsp> which is the list of all active SNOTEL sites. And the other goes to each states page from this one: <http://www.wcc.nrcs.usda.gov/snow/sntllist.html>. The harvester tests to see if a site is in the SNOTEL Sites table. If not it is added.

The data harvester runs from the list of sites in the sites table, and all SNOTEL sites in the country are in the table. It goes to an ftp site : <http://www.wcc.nrcs.usda.gov/ftpref/data/snow/snotel/cards/alaska/49t03s_all.txt>.

The SNOTEL database only contains metadata. So the harvester updates the sites table and the series catalog table so the data has the most recent Enddate and value count

The SNOTEL webservice , <http://drought.uwrl.usu.edu/SNOTEL/>, gets the request form the user and then queries the SNOTEL site to get the data it need. There are multiple files it has to access, one for the current water year and one for all other years. The code checks to see what files are needed and then splits up the request if required. Since we have to request all of the data from the file at a time a regular expression is created to filter out the requested data

USBR- Reservoir: flow through water one flow webservice and Metadata harvester.

Metadata harvester. This program uses a static list of sites in the code. Each site has its own variable code for each variable. So there is a class set up to hold the name of the site, The name used in the ftp url and all of the variables and their ids on the USBR website. This list was found I the source code of the USBR website at : <http://www.usbr.gov/uc/crsp/GetSiteInfo>. The code loops through this list to gather the most recent data for the series catalog table. The query can be limited by date so only the data requested is returned.

WOF webservice. This is set up to access the USBR Upper Colorado reservoir data website. It receives the request from the user and then generates a url based on the date times sent in the request and the data found in the static list of sites and their variable ids. the URL looks like this: [http://www.usbr.gov/uc/crsp/GetDataSet?l={URLSiteName}&c={VariableIDforsite}&strSDate=11-MAY-2012&strEDate=20-JUN-2012](http://www.usbr.gov/uc/crsp/GetDataSet?l=%7bURLSiteName%7d&c=%7bVariableIDforsite%7d&strSDate=11-MAY-2012&strEDate=20-JUN-2012) . this URL will create a .csv file, which is then accessed from the webservice and converted into a datavalues table.

SNODAS- Hold and serve database. Harvester and data loader and ODM Webservice. The dataloader accesses <ftp://ftp.nohrsc.nws.gov/pub/staff/nilsson/hucs/> which contains historical data from 1996-2010 in one format and 2011 data in another format. It then accesses <ftp://ftp.nohrsc.nws.gov/products/collaborators/> for the last month of 2012 data(only the last 30 days of data are saved here) I currently have a copy of all of the files loaded into the SNODAS huc 8, 10 and 12 databases. They are loaded into a regular ODM database. The code only saves the data if the site is in the database. And that list was created based on all the Hucs within and immediately bordering the UCRB.

The harvester is set up to read and unzip the 2012 data and load it into the correct database. It finds the last date of data loaded into the databases and load all file from that data to the current date.

Because the datavalues are actually saved to the database, a normal WOF webservice is used.

NCDC Harvester load and serve database, Harvester, WOF webservice

Only PRCP or precipitation data is loaded. Accesses an ftp site: [ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/all/{sitecode}.dly](ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/all/%7bsitecode%7d.dly) all of the data is saved to a datatable and then a filter is used to select the datavalues in the requested range.

Harvester loops through sites in the database to access the data. The stites table is created by using the sites within 100(50) miles of the UCRB border

Because the datavalues are actually saved to the database, a normal WOF webservice is used.

DataSummary is a c# .exe that is set up to generate levels 1-4 of data.

Command Line Arguments:

-l or- –level indicates what level you would like to run. You can select 0 or more at a time. entering nothing indicates you want to run all levels of data.  
 Possible entries: 1, 2, 3 and 4, separated by a semicolon. For accuracy the values should be in numerical order

-a or- –agency indicates the type of data you would like to run. You can select 0 or more at a time. Entering nothing indicates you want to run the program for all agencies.   
 Possible agencies: “NCDC-Weather”, “NRCS-SNOTEL”, “NRCS-SNOTEL” and “USBR-Reservoir”, separated by a semi-colon.

Home page. Handles command line arguments and calls a class dedicated to each level

getOriginalAgency. Prepares data for a WOF request

* Get list of Sites and webservices from L1Harvest List in the summary database.
* Loop through each of the entries.
  + Get start date by querying the Series catalog table of the ODM database by adding one day to the EndDateTime. If there is no entry in the Series catalog table set the start date to 1900-01-01. Set end date equal to ‘yesterday’.
  + Use webservice URL to determine whether it is 1\_0 or 1\_1. Based on result call another class

Level0Data1\_\*. WOF DataValues request. After the results are returned the code inserts the data into a datatable in the format of a datavalues table. After the datavalues are saved to the database indicated in the L1Harvest List the series catalog table is either updated or generated.

IndexVariables This class does the preliminary work to call the stored procedures CalcIndexVariables and IndexVariablesWatershed.

* IVtoTADxRef
* Get list of Sitetypes and series information from WatershedSeries in the summary database.

Subwatershed: loop through the sitetypes and call the stored procedure CalcIndexVariables  
Watershed: loop throught the sitetypes and call the stored procedure IndexVariablesWatershed.

IndexValues. Gathers the data needed to run the CalcIndexValue stored procedure.

Level 1 data., there are two classes that do the actual work for level1 data one for WOF 1.1 and one for WOF 1.0 webservices

We are using an ODM database with some alterations

QCL value indicated the level of data. E.g. level 2 data has a qualityControllevelcode of 2

Tables added: IVtoTADxRef, L1HarvestList, AggregateSeries, WatershedSeries, indexvaluesxref, UpstreamHucs

Stored Procedures: IndexValue, IndexVariable, L3Watershed, MonthlyAggregation, BiMonthlyAggregations, spbyquery, byemail, updatetableseriescatalog

Columns added: Sitetype in the Sites table

MonthlyDVMin in the Variabletable

DataTypes added: table DataValuesTable