Gnash Tutorial

Gnash

The tutorial below provides examples of identifying and extracting data using Gnash. It is intended to give the user a basic understanding of the application and some of its more frequently used commands.

The tutorial assumes that the files contained on the Centralised Dataset DVD distributed by the Authority have been unzipped onto the user's local hard drive. See the ReadMe.txt file in the root directory of the DVD for additional help on extracting the Gnash data files.

Identifying and extracting meter data

Open Gnash by running the Gnash.exe file contained in the HalfHourly directory in the Centralised Dataset. This will open a window similar to a standard DOS window with a "Gnash :" prompt where commands can be entered.

Identifying data series names

Gnash provides access to active and reactive meter data by grid exit point, generation by grid injection point, HVDC flows, bids and offers, final prices, reserve prices and binding constraints information.

- Meter data is provided in kW and kVar units for active and reactive data respectively.
- Price information is in $/MWh units.

The various series containing the data are named using a tree structure.

The hide command can be used to hide series or groups of series from all output produced by Gnash. This is useful if you are only interested in meter data for example, and not prices or bids and offers. The command ***hide ~bid*** hides all bids data. The command ***unhide ~bid*** would show the bids data again. ***Unhide ~*** shows all data previously hidden using the hide command.

The following is an example of the names of data series relating to the Kaitaia grid exit point obtained using the ***names ~kaitaia~*** command (after bids have been hidden):

```
30216 NI.Kaitaia                           Kaitaia (Bay of Islands).
28648 NI.Kaitaia.110KV.Price               Kaitaia (Northland) 110KV Putative Price.
28647 NI.Kaitaia.110KV.Price.Price_Run_Time Kaitaia (Northland) 110KV Putative Price.
28985 NI.Kaitaia.110KV.Price.TimeStamp     Kaitaia (Northland) 110KV Putative Price.
29229 NI.Kaitaia.11KV                      Kaitaia (Northland) 11KV Transformer 2.
29227 NI.Kaitaia.33KV                      Kaitaia (Northland) 33KV Transformer 4.
```

```
27790 NI.Kaitaia.33KV.Price                 Kaitaia (Northland) 33KV Transformer 4 Putative Price.
28307 NI.Kaitaia.33KV.Price.PRICE_RUN_TIME  Kaitaia (Northland) 33KV Transformer 4 Putative Price.
28027 NI.Kaitaia.33KV.Price.TimeStamp       Kaitaia (Northland) 33KV Transformer 4 Putative Price.
30550 NI.Kaitaia.33KV.Reactive.Nett         Kaitaia (Northland) 33KV Transformer 4 Reactive, nett.
30569 NI.Kaitaia.33KV.Reactive.Nett.v2      Kaitaia (Northland) 33KV Transformer 4 Reactive,nett,v2
30568 NI.Kaitaia.33KV.v2                    Kaitaia (Northland) 33KV Transformer 4, v2.
29669 NI.Kaitaia.Local                      Kaitaia (Northland) Local Service.
29228 NI.Kaitaia.Pukenui                    Kaitaia (Northland) 50KV to Pukenui.
29230 NI.Kaitaia.Sum                        Kaitaia (Northland) "Sum".
```

The names command accepts the ~ character as a “wild card”. The above list would also have been generated if the command ***names ni.kait~*** was used for example.

The above output shows a unique internal reference number for each series (e.g. 29229), the names of the series (NI.KAITAIA.11KV), and a descriptive title for the series (“Kaitaia (Northland) 11KV Transformer 2.”).

Each individual series is a collection of historical data files covering different date ranges. In many cases the series will only contain data for limited periods of time due to grid configuration changes, or changes in meter data collection or processing.

Grid exit points

For grid exit points the naming convention used in Gnash is to prefix each series name with an Island reference (i.e. either NI for North Island or SI for South Island), followed by the substation name, any additional information (bus name for example), and where the data is not active meter data, the type of data.

Each part of the name is separated by a period. Hence the NI.KAITAIA.33KV.REACTIVE series contains reactive meter data for the 33KV bus at Kaitaia.

There are some special cases where the prefix for grid exit point data is not NI or SI. For example, there are a number of pre-calculated sums provided with the raw meter data that are prefixed with a SUM code.

Tiwai Point aluminium smelter loads are prefixed with the code TY. To list the various data series containing data relating to the Tiwai smelter, use the command ***names ty~*** .

Generation meter data

Series containing meter data associated with generation connected to the grid (either through Grid Injection Points or the point at which the generator connects to the local lines network) are prefixed with a GEN reference and have the general structure

GEN.GenerationType.GenerationLocation.GenerationName

For example, GEN.Wind.Wellington.Brooklyn is the series containing meter data for the Brooklyn wind turbine in Wellington.

Other data series

Price data has a ‘.Price’ suffix. Offer data has an ‘.Offer’ suffix and bid data a ‘.Bid’ suffix with the bids and offers data also having a secondary ‘Power’, ‘Price’ or ‘Ramp’ identifier in the series name where applicable.

Checking series for data

For this example we are looking for active meter data for the Kaitaia series identified above, so we are interested in the NI.KAITAIA, NI.KAITAIA.11KV and NI.KAITAIA.33KV series. The various NI.KAITAIA.33KV.PRICE and NI.KAITAIA.33KV.REACTIVE series contain price and reactive data respectively.

The LOCAL, PUKENUI and SUM series are also potential sources of data so we need to take a look at what is in each.

The ***stats*** command can be used to examine the data in each series at a high level. Because we are interested in data specifically between 1997 and 2009 we can use the command ***stats ni.kaitaia~ for 1997-2009*** to have a look at what is available (the ***stats*** command will accept more specific dates in a 30/6/2009 or a 30/6/9 format if needed).

```
   Run Name                                  Date Span              DaysHolesZombi Values
30477 NI.Kaitaia                            1/ 1/1997-31/12/2009 4748    0    0 227904
28911 NI.Kaitaia.110KV.Price               21/ 7/2009-19/ 8/2009   30    0    0   1417
28910 NI.Kaitaia.110KV.Price.Price_Run_Time 21/ 7/2009-31/ 7/2009   11    0    0    505
29247 NI.Kaitaia.110KV.Price.TimeStamp      1/ 8/2009-19/ 8/2009   19    0    0    912
29491 NI.Kaitaia.11KV                      A void.               No grist to grind.
29489 NI.Kaitaia.33KV                       1/ 1/1997-31/12/2009 4748    0    0 227904
28055 NI.Kaitaia.33KV.Price                 1/ 1/1997-31/12/2009 4748    0    0 227904
28572 NI.Kaitaia.33KV.Price.PRICE_RUN_TIME  1/ 1/2004-31/ 7/2009 1705  334    0  81844
28292 NI.Kaitaia.33KV.Price.TimeStamp       1/ 1/1997-31/12/2009 2923 1825    0 140298
30809 NI.Kaitaia.33KV.Reactive.Nett         1/ 1/1997-31/12/2009 4748    0    0 227904
30828 NI.Kaitaia.33KV.Reactive.Nett.v2      1/ 1/2000-30/ 6/2004 1463  180    0  70224
30827 NI.Kaitaia.33KV.v2                    1/ 1/2000-31/12/2000  365    1    0  17522
29931 NI.Kaitaia.Local                     A void.               No grist to grind.
29490 NI.Kaitaia.Pukenui                   A void.               No grist to grind.
29492 NI.Kaitaia.Sum                       A void.               No grist to grind.
```

The above output has been truncated at the right of the page for this example. The actual output contains additional information for those series where data is present

In this case the output shows that only the NI.KAITAIA, NI.KAITAIA.33KV, NI.KAITAIA.33KV.PRICE, and NI.KAITAIA.33KV.REACTIVE series have data for the period we are looking at. The NI.KAITAIA series is a summary series which includes the 33KV data. It also includes data for the 11KV supply when that was active.

In this case we only need the data from the NI.KAIATIA series for our analysis of total North Isthmus demand. (Using the command ***stats ni.kaitaia~*** shows that the 11KV, LOCAL, PUKENUI and SUM series contain meter data for a mix of periods between 1984 and 1996.)

Data for some substations is available at both a summary level and for individual buses at that same station. Care needs to be taken not to double-count the load at that station by including both series. This issue can usually be identified easily as there will be series containing individual bus data such as NI.KAITAIA.33KV above, and a parent series NI.KAITAIA also containing data. Similarly, LOCAL loads are sometimes associated with substation facilities and while this demand is reported as a separate series it is normally already included in other series (NI.OTAHUHU and NI.OTAHUHU.LOCAL are an example where the LOCAL load is already included in the substation total).

Collating regional data

To create a total for the North Isthmus region it is necessary to identify each relevant substation, and check the specific series that data is available for using the ***stats*** command e.g. ***stats ni.wellsford~ for 1997-2009*** and ***stats ni.kaikohe~ for 1997-2009*** etc.

Doing so yields the following list of series containing demand data for the North Isthmus region during the 1997 to 2009 period:

NI.ALBANY, NI.BREAMBAY, NI.DARGAVILLE, NI.HENDERSON, NI.HEPBURN.AUCKLAND, NI.HEPBURN.WAITEMATA, NI.KENSINGTON, NI.KAIKOHE, NI.KAITAIA, NI.MARSDEN, NI.MAUNGATAPERE, NI.MAUNGATUROTO, NI.SILVERDALE.EXPORT and NI.WELLSFORD

Summing and grouping data

Series can be added together (or have other calculations applied to them) by using the ***calculate*** command. This command creates a new variable that can be used in subsequent calculations. To add together the above series use the command

calculate SUM.NORTHISTHMUS = NI.ALBANY + NI.BREAMBAY + NI.DARGAVILLE + NI.HENDERSON + NI.HEPBURN.AUCKLAND + NI.HEPBURN.WAITEMATA + NI.KENSINGTON + NI.KAIKOHE + NI.KAITAIA + NI.MARSDEN + NI.MAUNGATAPERE + NI.MAUNGATUROTO + NI.SILVERDALE.EXPORT + NI.WELLSFORD

This will create a new series called SUM.NORTHISTHMUS (note that the commands in Gnash are not case sensitive).

Unfortunately, if we now use the command ***stats sum.northisthmus for 1997-2009*** we will find that the series is empty. This illustrates an important difference between series containing “zeroes” and missing data (null data). An attempt to add a series containing data to a missing series will result in null data. So while a series may exist, if it has no data in the period you are analysing (as opposed to having data but it being all zeroes) then any calculation including it will produce a null result for that period. In the above case, while each of the individual series has at least some data during the 1997-2009 period, there was no day where **all** of the series had data.

A useful tool for looking for gaps in data is the ***presence*** command. The ***presence*** command provides a visual description of the existence of data in a specified series. Where the series requested was defined by a ***calculate*** command, the ***presence*** command will also show the existence of data for all of the series directly referred to in that calculation.

The following text shows a subset of the output produced by the command ***presence sum.northisthmus for 2003*** (assuming that the above ***calculate*** command has been run to define SUM.NORTHISTHMUS first).

```
Entirely absent for Wednesday 1/ 1/2003 - Wednesday 31/12/2003
NI.Marsden           <--> Marsden (Northland).
      14 runs to check data presence for Wednesday 1/ 1/2003 - Wednesday 31/12/2003: 365 d
                SUM.NORTHISTHMUS <--> NI.ALBANY + NI.BREAMBAY + NI.DARGAVILLE + NI.HENDERS…
                |NI.Albany <--> Albany (North Auckland).
                ||NI.BreamBay <--> Bream Bay (Auckland).
                |||NI.Dargaville <--> Dargaville (Northland).
                ||||NI.Henderson <--> Henderson (Auckland).
                |||||NI.Hepburn.Auckland <--> Hepburn Road to Auckland city (not Waitemata…
                |||||!NI.Hepburn.Waitemata <--> Hepburn Road to Waitemata (not Auckland).
                |||||!|NI.Kensington <--> Kensington (Whangarei - Northland).
                |||||!||NI.Kaikohe <--> Kaikohe (Northland).
                |||||!|||NI.Kaitaia <--> Kaitaia (Bay of Islands).
                |||||!||||NI.Maungatapere <--> Maungatapere (Northland, west of Whangarei)
                |||||!|||||NI.Maungaturoto <--> Maungaturoto (Northland).
                |||||!|||||!NI.Silverdale.Export <--> Silverdale (Orewa - North Auckland).
                |||||!|||||!|NI.Wellsford <--> Wellsford (Waitemata).
                |||||!|||||!||
                vvvvvvvvvvvvvv
Wed  1/ 1/2003:  54343444444 4
~same        4   54343444444 4
Mon  6/ 1/2003:  54344444444 4
~same        4   54344444444 4
Sat 11/ 1/2003:  54343444444 4
     ……
     ……
Tue 28/10/2003:  54344444444 4
~same        3   54344444444 4
Sat  1/11/2003:  54343444444o4
Sun  2/11/2003:  54343444444o4
Mon  3/11/2003:  54344444444o4
     ……
     ……
Sun 30/11/2003:  54343444444o4
Mon  1/12/2003:  54344444444o4
~same        3   54344444444o4
Fri  5/12/2003:  5434444444444
Sat  6/12/2003:  5434444444444
Sun  7/12/2003:  5434344444444
     ……
     ……
Tue 23/12/2003:  5434444444444
Wed 24/12/2003:  5434344444444
~same        7   5434344444444
```

(note that data for some dates has been removed from this example output)

The individual figures in the printout show the magnitude of the maximum value held for the series for that day, as a power of 10 (e.g. a value of 4 indicates that the maximum load for the day was over 10000 kW but under 100000 kW). Zeroes show as a ”o”, as distinct to a “0” which indicates a value between 1kW and 9kW. A space is shown where data is completely missing for that day, and a “-“ indicates that the maximum value is negative. Price data and pre-1996 meter data may also show a “?” where data is available for the day but at least one half hour is missing data, and a “z” where data is marked as missing for the entire day in the source files[1].

In the 2003 year, NI.MARSDEN does not contain any data at all (noted at the very top of the output). As a result, SUM.NORTHISTHMUS is shown as containing null values in the body of the output. The output also shows that NI.SILVERDALE.EXPORT contains no data until 1 November 2003, and does not contain any non-zero data until 5 December 2003.

[1] This is subtly different from a “space” missing value where the source files don’t include any information at all for that day. In this case the data is actually flagged as missing for the individual half hours recorded for that day in the source files.

There are two solutions to this problem. The ***mergeadd*** function can be used to merge together datasets without discarding values in periods where data is absent in some series. ***Mergeadd*** uses commas to separate the data series as it is specified in a functional form rather than as an equation. If the command

calculate SUM.NORTHISTHMUS = mergeadd(NI.ALBANY, NI.BREAMBAY , NI.DARGAVILLE , NI.HENDERSON , NI.HEPBURN.AUCKLAND , NI.HEPBURN.WAITEMATA , NI.KENSINGTON , NI.KAIKOHE , NI.KAITAIA , NI.MARSDEN , NI.MAUNGATAPERE , NI.MAUNGATUROTO, NI.SILVERDALE.EXPORT, NI.WELLSFORD)

is used then the SUM.NORTHISTHMUS series will now have data for the entire period from 1997 to 2009.

It is important to recognise that the summing of individual series in this way is only valid for the period where the existence of data for individual series has been checked (i.e. 1997 to 2009). The SUM.NORTHISTHMUS series will also contain data and give results for the period 1986 to 1996, but it would be necessary to check whether there are any other relevant series within the region that need to be included across that earlier time period. The Authority is still in the process of organising the data in the Centralised Dataset. While most of the regions now contain series that can consistently be used from 1986 onwards, there are still some areas where historical system configuration changes are being investigated. Therefore summary level series prior to the start of the wholesale market on 1 October 1996 should be treated with caution.

An alternative approach to dealing with null data is to use the ***zpad*** function. This function will take an individual series and pad the series with zeroes where data does not exist. In the above example the NI.MARSDEN, NI.HEPBURN.WAITEMATA and NI.SILVERDALE.EXPORT series all contain periods where data is missing between 1997 and 2009.

An example command using this function is:

calculate SUM.NORTHISTHMUS = NI.ALBANY + NI.BREAMBAY + NI.DARGAVILLE + NI.HENDERSON + NI.HEPBURN.AUCKLAND + zpad(NI.HEPBURN.WAITEMATA) + NI.KENSINGTON + NI.KAIKOHE + NI.KAITAIA + zpad(NI.MARSDEN) + NI.MAUNGATAPERE + NI.MAUNGATUROTO + zpad(NI.SILVERDALE.EXPORT) + NI.WELLSFORD

Before using ***zpad*** or ***mergeadd*** it is worthwhile checking that the missing data is not represented in another series that you have not included into your calculation (using the ***stats*** or ***names*** command).

Grid Injection vs. Grid Export data

The sign of meter data can vary depending on whether energy is flowing into or out-of the grid. The direction indicated by positive meter values depends on whether the metering is carried out at a grid exit point or a grid injection point.

As covered above, meter data at grid injection points is prefixed with the code GEN. Meter data in the GEN series has a positive sign when energy is being injected on to the grid.

Meter data in the grid exit point series has a positive sign when energy is being exported from the grid. However, there are some buses classed as grid exit points that inject into the grid from time to time. Generally this occurs where there is significant embedded generation connected to the local network behind the grid exit point. In some data series the injection data is included and shown as a negative number. In other data series a 0 is shown for those half hours where energy is being injected into the grid. In these cases there will be a separate corresponding data series containing the injection data for that grid exit point.

To avoid subtracting grid injection information from demand (where the injection is associated with grid exit points) it is necessary to remove the negative data from any summations. The ***pos*** function can be used to do this for each series. While most of the series do not contain negatives (the ***stats*** command can be used to check maximum and minimum values), the Authority normally applies the ***pos*** function to all series where multiple grid exit points are being summed to calculate total demand for an area.

For the North Isthmus example above, total demand excluding any injection at grid exit points can be calculated using the following command:

calculate Sum.NorthIsthmusPost96 = pos(NI.ALBANY) + pos(NI.BREAMBAY) + pos(NI.DARGAVILLE) + pos(NI.HENDERSON) + pos(NI.HEPBURN.AUCKLAND) + pos(zpad(NI.HEPBURN.WAITEMATA)) + pos(NI.KENSINGTON) + pos(NI.KAIKOHE) + pos(NI.KAITAIA) + pos(zpad(NI.MARSDEN)) + pos(NI.MAUNGATAPERE) + pos(NI.MAUNGATUROTO) + pos(zpad(NI.SILVERDALE.EXPORT)) + pos(NI.WELLSFORD)

Outputting data to text files

Once the desired data series have been identified or calculated the next step is usually to export the data to a file for further analysis, graphing etc.

The ***dump*** command exports data in a comma delimited text format to a specified file.

To export the Sum.NorthIsthmusPost96 series using the ***dump*** command
dump Sum.NorthIsthmusPost96 for 1997-2009 to NorthIsthmusPost96.csv

This will export the meter data to the file NorthIsthmusPost96.csv with a number of additional columns of information.

By default these are:

Date Half-hour Days since 31/12/1899 Day Class Daylight Saving State Solar Azimuth Solar Altitude

The date and half hour are New Zealand standard time – i.e. the dates and half hours reflect daylight saving changes, so there is a day with 50 half hours and a day with 46 half hours in each year.

“Days since 31/12/1889” is a simple count of days with fractions indicating half hour steps during the day.

Day Class is a code that indicates the type of day – whether it is a weekday/weekend or holiday. Day codes 1-5 = Monday to Friday, codes 16-17 = Saturday and Sunday, codes 30-45 = Holidays[2]. Regional holidays are not shown.

Daylight Savings State shows 1 for daylight savings period, 2 for no daylight savings period.

Solar Azimuth is the bearing of the sun with respect to geographical north.

Solar Altitude shows the angle of the sun to horizontal (useful for analysis where day/night timing differences are an issue).

The default columns exported by Gnash can be altered using the ***set*** command. To remove the solar altitude for example use the ***set DUMP.AUX.SOLAR.ALTITUDE = F*** command. A full list of the variables that can be changed using the ***set*** command can be examined by simply entering ***set*** by itself.

Batch files and storing calculation definitions

Regularly used expressions or summaries can be stored and run from text files using the @ or ***xeq*** commands.

The format of the text files is exactly the same as the text that would be entered into Gnash directly. Text can be entered directly into files using a text editor, pasted from other text files such as Gnash.Trail.txt and Gnash.Gnashed.txt (a log of what has been typed into Gnash during the current session) or pasted from the Gnash window if your version of Windows allows.

The Authority has created a set of region total calculations and has included it within the Centralised Dataset files to provide a starting point for grouping grid exit points. The region definitions used are consistent with region definitions used in the 2008 Statement of Opportunities and should be treated as being only valid for dates after 1/10/1996 (i.e. the commencement of the wholesale electricity market).

The regional total calculations are contained in the file Regions.txt. Use the command ***@Regions.txt*** to execute the various ***calculate*** commands that are contained in the file. The relevant series can then be exported to text files if desired[3].

If you wish to run a number of commands automatically when Gnash is started (useful for customising inputs/outputs or including standard calculations such as the region totals above) then create a new text file called “Gnash.awake.txt” in the same directory that the Gnash.exe file is located in. Gnash looks for this file on start-up and if the file exists will execute any commands contained within it.

---

[2] Good Friday = 30, Easter Monday = 31, ANZAC Day = 32, Waitangi Day = 33, Queens Birthday = 34, Labour Day = 35, Christmas Day = 36, Boxing Day = 37, Christmas Day Holiday (if different to Christmas Day) = 38, Boxing Day Holiday (if different to Boxing Day) = 39, New Years Day = 40, Day after New Years Day = 41, New Years Day Holiday (if different to New Years Day) =42, Day after New Years Day Holiday (if different to Day after New Years Day)=43, Week days between Christmas and New Years (that are not public holidays) = 44, Week days in the remainder of the week following New Years (that are not public holidays) = 45.

[3] The various summary series created by executing the calculations contained in the Regions.txt file have been prefixed with the code **REG** to distinguish them from the base series. They can be exported to a text file by entering the command ***dump reg.~ for 1997-2009 to RegionTots.csv***

A good example is if you don't like Gnash closing down when an empty line is entered (the default), then putting the ***set stoponblank = f*** command in the Gnash.awake.txt file will require you to type the ***stop*** command to exit Gnash.

Comments can be added to the end of statements in text files (or added to commands directly typed into Gnash) using the ***%*** character to mark the start of comment text. Any comment text at the end of a ***calculate*** command will be used as the descriptive title for the new series created by the statement.

Extension : Importing the data into MATLAB and plotting

Depending on the version of Excel you use, the length of the text files exported by Gnash may be too long to fully import into Microsoft Excel if the period selected is longer than about 3 ½ years. Versions of Excel prior to Excel 2007 will only accept a maximum of 65536 lines of data, with the earlier versions accepting even fewer lines. Excel can also only graph a maximum of 32000 data points. With each non-leap year containing 17520 half hours, analysis of more than a few years' data is usually best carried out using tools better suited to handling large quantities of data.

The Authority has settled on MATLAB as a standard development tool for a number of its modelling activities. MATLAB is capable of handling large data sets and has a flexible and powerful scripting language[4].

Files can be imported on an ad-hoc basis into MATLAB using ***Import Data*** under the MATLAB ***File*** menu item. However, the resulting data array will need to be matched to the separate text array that MATLAB creates containing the series names and descriptions. This can be a fiddly process for files that contain several series and it is generally easier to set up scripts that import the text files into a more convenient data structure. The Authority has created a script for this purpose called ***Gnash.m*** that has been provided in the ***HalfHourly/Auxiliaries*** directory in the Centralised Dataset.

Earlier versions of MATLAB are unable to cope with directory names containing spaces. It is recommended that ***Gnash.m*** and any text files that you wish to import into MATLAB are copied to a directory where the full path name will not contain spaces. Ensure that the directory containing ***Gnash.m*** is included in the MATLAB search path (see ***Set Path*** under the MATLAB ***File*** menu item).

For this example we created a new directory 'C: \CDS' and copied the file ***NorthIsthmusPost96.csv*** created in the above tutorial into it.

Open MATLAB.

The filename to be imported into MATLAB by the ***Gnash.m*** script needs to be identified by entering the following command at the >> prompt (note that MATLAB is case sensitive):

GnashFile = 'C:\CDS\ NorthIsthmusPost96.csv'

[4] An alternative to MATLAB is the open source application OCTAVE. OCTAVE has limited compatibility with MATLAB scripts so it is possible that some changes will need to be made to Authority-provided scripts in order for them to work.

Alternatively the filename without the full path name can be specified provided that the current directory defined in MATLAB is the one containing the file (see the MATLAB command ***cd***).

The file specified can now be imported by simply entering the command ***Gnash*** at the >> prompt**.**

The import process may take several seconds depending on the size of the file, so if it appears to halt mid-run it is likely that the script is still running.

The script will automatically convert the date number code used by Gnash into the standard MATLAB date numbering system.

A number of variables are produced by the script from the imported data. The full set of imported data can be found in a cell called ***GnashData***. This is in a format where each information and data series is a separate item in a 2-dimensional cell. The names and titles for each column of data in ***GnashData*** are contained in the ***GnashNames*** and ***GnashTitles*** variables.

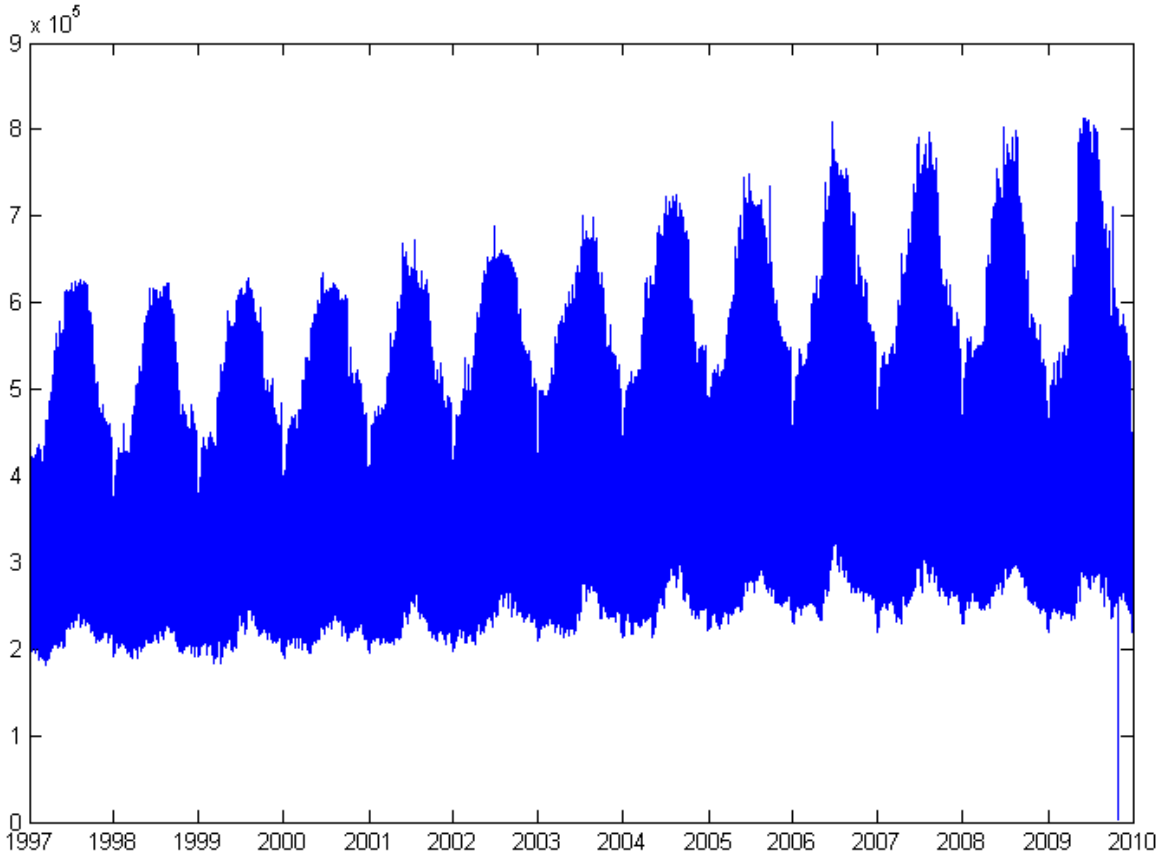
The variables ***GnashRows*** and ***GnashCols*** contain the number of rows of data and the number of information/data columns in ***GnashData*** respectively.

The script also provides the data in a structured format similar to the tree structure used by Gnash. The structure variable ***Aux*** contains the information variables from the text file. By default these will be ***Date***, ***HHn***, ***DayClock***, ***DayClassCode, DayLightSaving*** and ***Solar*** (azimuth and altitude) as covered in the ***Outputting data to files*** section above. These series can be accessed using the names ***Aux.Date***, ***Aux.HHn*** etc.

The data itself will be in a variable with the name of the series prefix (in this case ***Sum***). The individual data series can be referred to directly using the same names as the Gnash series names. An exception is where the portion of the Gnash name starts with a number. In those cases that portion of the name is prefixed with a 'v' by the import script – e.g. NI.KAITAIA.33KV would show as NI.KAITAIA.v33KV in the MATLAB structure, had it been imported.

For the North Isthmus example above, the data series from 1997 to 2009 can be plotted (with dates on the X axis) using the MATLAB command
plot(Aux.DayClock , Sum.NorthIsthmusPost96)
followed by the ***datetick*** command to display the date numbers in a date format.

This will result in the following graph being displayed:



Titles and axis names can then be added using the appropriate MATLAB commands or the graph interface (the dip in late 2009 is correct and shows the outage that occurred in the Auckland and North Isthmus regions on the morning of 30 October).

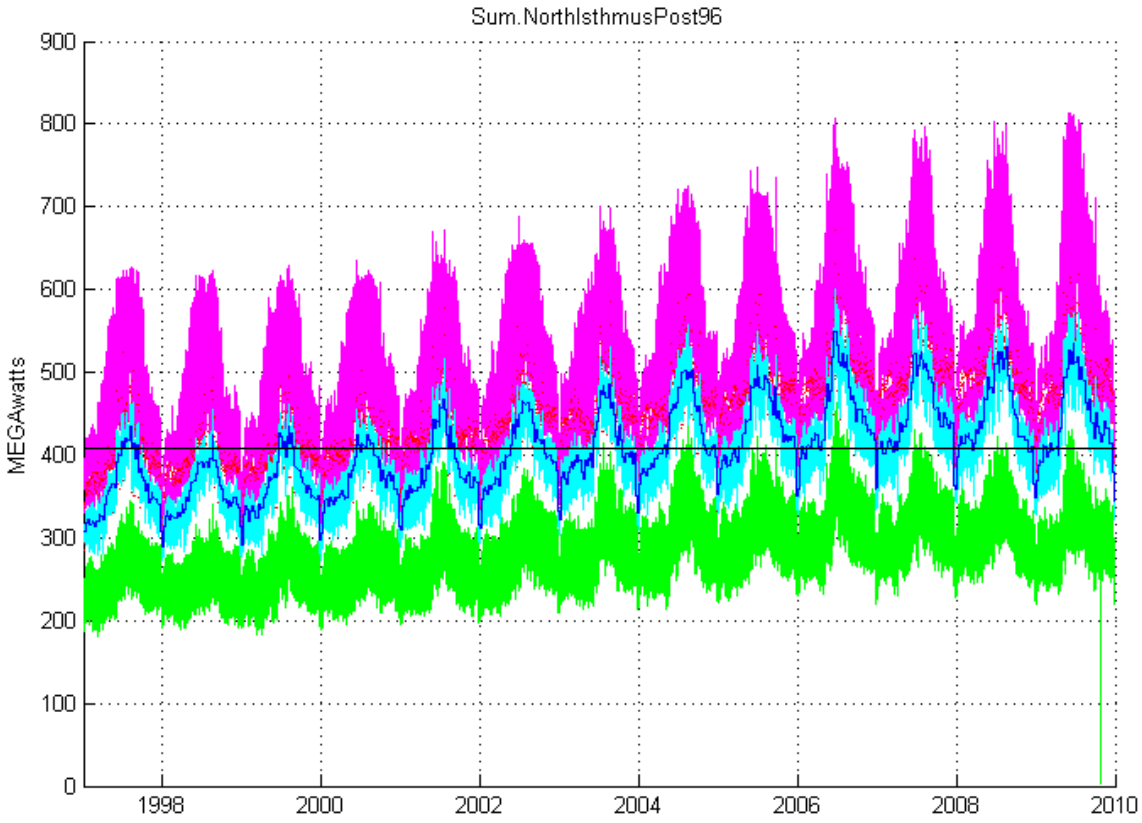
An alternative to using the standard MATLAB plotting functions is to use the GnashPlot utility that has also been provided in the ***HalfHourly/Auxiliaries*** directory in the Centralised Dataset.

To use GnashPlot the directory containing both the GnashPlot.m and DayPlot.m files needs to be included in the MATLAB path. GnashPlot also relies on some recently introduced standard MATLAB functions so it may not work with some older versions of MATLAB.

Once data has been imported using the Gnash utility, just type **GnashPlot** at the MATLAB >> prompt.

GnashPlot will produce a plot for each series that has been imported. The colours in the graph indicate various statistics associated with the daily values in the series. Vertical purple lines show the upper quartile range for the day, vertical green lines show the lower quartile range for the day, the daily median is shown by short horizontal red lines, and the daily mean by a light-blue line. The weekly mean is shown using a dark-blue horizontal line and the mean for the entire series is indicated by a horizontal black line.

The following plot shows the output of GnashPlot for the Sum.NorthIsthmusPost96 series from the above exercise. Some of the more detailed statistics are obscured because of the number of data points in the diagram below, but these would be viewable in MATLAB by zooming into a shorter time period.


Sum.NorthIsthmusPost96
MEGAwatts
900
800
700
600
500
400
300
200
100
0
1998
2000
2002
2004
2006
2008
2010