# Un tutoriel pour libhydro (version 0.2 - maj 20130830)

## Le module core.sitehydro: les sites, les stations et les capteurs

**Usage normal:**

In [1]:
```
from libhydro.core import sitehydro

s = sitehydro.Sitehydro('O0334010', libelle=u'La Garonne à Toulouse')
print(s)
```

```
Site REEL O0334010::La Garonne à Toulouse [0 station]
```

In [2]:
```
s.stations = [
    sitehydro.Stationhydro('O033401001', libelle=u'La Garonne à Toulouse - échell
    sitehydro.Stationhydro('O033401001', libelle=u'La Garonne à Toulouse - échell
]
s.typesite = 'SOURCE'
print(s)
for t in s.stations: print t
```

```
Site SOURCE O0334010::La Garonne à Toulouse [2 stations]
Station LIMNI O033401001::La Garonne à Toulouse - échelle principale [0 capteur]
Station LIMNI O033401001::La Garonne à Toulouse - échelle de secours [0 capteur]
```

In [3]:
```
c = sitehydro.Capteur('O03340100201', 'H', 'radar')
print(c)
```

```
Capteur H O03340100201::radar
```

**Usage libre avec strict=False:**

In [4]:
```
s = sitehydro.Sitehydro('site 10', stations=['station1', 'station2'], strict=Fals
print s
# il faut au minimum un code...
s = sitehydro.Sitehydro(10, strict=False)
print(s)
```

```
Site REEL site 10::<sans libelle> [2 stations]
Site REEL 10::<sans libelle> [0 station]
```

In [5]:
```
c = sitehydro.Capteur(2, 'vitesse', strict=False)
print(c)
```

```
Capteur vitesse 2::<sans libelle>
```

## Le module core.obshydro

Les obshydro.Observations sont des pandas.DataFrame, dont l'index est une série de timestamp. La seule colonne

obligatoire est la colonne résultat.

**La classe Serie s'appuie sur les classe Observation et Observations:**

In [39]:
```
from libhydro.core import sitehydro
from libhydro.core import obshydro

r = obshydro.Serie(
    entite=sitehydro.Sitehydro('O0334010', libelle=u'La Garonne à Toulouse'),
    grandeur='H',
    observations=obshydro.Observations(
        obshydro.Observation('2012-10-05 08:33+00', 10),
        obshydro.Observation('2012-10-05 09:33Z', 20),
        obshydro.Observation('2012-10-05 12:33', 30),
        obshydro.Observation('2012-10-06 11:33', 40),
        obshydro.Observation('2012-10-07 11:33', 50),
        obshydro.Observation('2012-10-08 11:33', 60)

    )
)
print('ATTENTION, heure locale => TU !')
print(r)
```

```
ATTENTION, heure locale => TU !
Serie H sur le Site REEL O0334010::La Garonne à Toulouse [0 station]
Statut 0::sans validation
--------------------------------------------------------------------------------
Observations:
                     res  mth  qal   cnt
dte
2012-10-05 08:33:00   10    0   16  True
2012-10-05 09:33:00   20    0   16  True
2012-10-05 10:33:00   30    0   16  True
2012-10-06 09:33:00   40    0   16  True
2012-10-07 09:33:00   50    0   16  True
2012-10-08 09:33:00   60    0   16  True
6 values
```

**Extraction de valeurs**

In [38]:
```
# récupérer une pandas.Serie ne contenat que la colonne 'res'
print(r.observations.res) # idem r.observations['res']
print('-' * 70)
# avec ix on peux récupérer l'observation d'un timestamp précis...
print(r.observations.ix['2012-10-05 10:33+00'])
print('-' * 70)
# ... ou imprécis
print(r.observations.ix['2012-10-05'].res)
```

```
dte
2012-10-05 08:33:00    10
2012-10-05 09:33:00    20
2012-10-05 10:33:00    30
2012-10-06 09:33:00    10
2012-10-07 09:33:00    20
2012-10-08 09:33:00    30
Name: res, dtype: float64
----------------------------------------------------------------------
res      30
mth       0
qal      16
cnt    True
Name: 2012-10-05 10:33:00, dtype: object
----------------------------------------------------------------------
dte
2012-10-05 08:33:00    10
2012-10-05 09:33:00    20
2012-10-05 10:33:00    30
Name: res, dtype: float64
```

**Slicing**

In [40]:
```
r.observations['2012-10-05 10:33+00':]
```

Out[40]:

| | res | mth | qal | cnt |
|---|---|---|---|---|
| **dte** | | | | |
| **2012-10-05 10:33:00** | 30 | 0 | 16 | True |
| **2012-10-06 09:33:00** | 40 | 0 | 16 | True |
| **2012-10-07 09:33:00** | 50 | 0 | 16 | True |
| **2012-10-08 09:33:00** | 60 | 0 | 16 | True |

In [41]:
```
r.observations[:'2012-10-05 11:33+00']
```

Out[41]:

| | res | mth | qal | cnt |
|---|---|---|---|---|
| **dte** | | | | |
| **2012-10-05 08:33:00** | 10 | 0 | 16 | True |
| **2012-10-05 09:33:00** | 20 | 0 | 16 | True |
| **2012-10-05 10:33:00** | 30 | 0 | 16 | True |

**Concaténation**

In [49]:
```
obshydro.Observations.concat(r.observations, r.observations)
```

Out[49]:

| | res | mth | qal | cnt |
|---|---|---|---|---|
| **dte** | | | | |
| **2012-10-05 08:33:00** | 10 | 0 | 16 | True |
| **2012-10-05 09:33:00** | 20 | 0 | 16 | True |
| **2012-10-05 10:33:00** | 30 | 0 | 16 | True |
| **2012-10-06 09:33:00** | 40 | 0 | 16 | True |
| **2012-10-07 09:33:00** | 50 | 0 | 16 | True |
| **2012-10-08 09:33:00** | 60 | 0 | 16 | True |
| **2012-10-05 08:33:00** | 10 | 0 | 16 | True |
| **2012-10-05 09:33:00** | 20 | 0 | 16 | True |
| **2012-10-05 10:33:00** | 30 | 0 | 16 | True |
| **2012-10-06 09:33:00** | 40 | 0 | 16 | True |
| **2012-10-07 09:33:00** | 50 | 0 | 16 | True |
| **2012-10-08 09:33:00** | 60 | 0 | 16 | True |

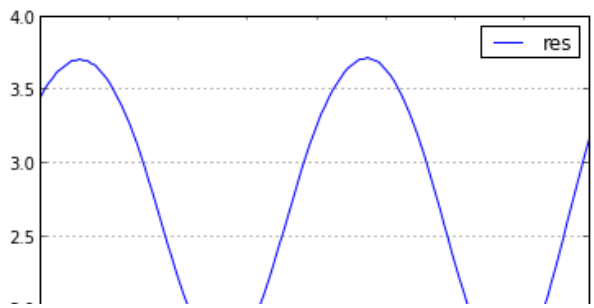
**Une obshydro.Serie à partir du convertisseur SHOM:**

In [50]:
```
from libhydro.conv import shom

r = shom.serie_from_hfs('../../../test/data/shom/LOCMARIAQUER.hfs')
print(r)
r.observations.plot()
```

```
Serie H sur la Station LIMNI <sans code>::LOCMARIAQUER [0 capteur]
Statut 0::sans validation
--------------------------------------------------------------------------------
Observations:
                     res
dte
2013-01-23 00:00:00  3.46
2013-01-23 00:10:00  3.51
2013-01-23 00:20:00  3.55
2013-01-23 00:30:00  3.58
2013-01-23 00:40:00  3.62
2013-01-23 00:50:00  3.64
2013-01-23 01:00:00  3.66
2013-01-23 01:10:00  3.68
2013-01-23 01:20:00  3.70
2013-01-23 01:30:00  3.70
2013-01-23 01:40:00  3.71
2013-01-23 01:50:00  3.70
2013-01-23 02:00:00  3.70
2013-01-23 02:10:00  3.68
2013-01-23 02:20:00  3.67
...
2013-01-23 21:30:00  1.85
2013-01-23 21:40:00  1.92
2013-01-23 21:50:00  2.01
2013-01-23 22:00:00  2.09
2013-01-23 22:10:00  2.19
2013-01-23 22:20:00  2.28
2013-01-23 22:30:00  2.38
2013-01-23 22:40:00  2.48
2013-01-23 22:50:00  2.59
2013-01-23 23:00:00  2.69
2013-01-23 23:10:00  2.79
2013-01-23 23:20:00  2.89
2013-01-23 23:30:00  2.99
2013-01-23 23:40:00  3.09
2013-01-23 23:50:00  3.18
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 144 entries, 2013-01-23 00:00:00 to 2013-01-23 23:50:00
Data columns (total 1 columns):
res    144  non-null values
dtypes: float64(1)
```

Out[50]: <matplotlib.axes.AxesSubplot at 0x359e050>



# Le module core.simulation

Les simulation.Previsions sont des pandas.Series avec un double index (hiearchical index): une date et une probabilité de prévision. En dérogation au modèle de données, les résultats (min, moy ,max) sont traités comme des prévisions de probabilité respective (0, 50, 100).

### Instancier une simulation

In [66]:
```
from libhydro.core import simulation, sitehydro

u = simulation.Simulation(
    entite=sitehydro.Stationhydro(code='O034552001'),
    grandeur='H',
    previsions=simulation.Previsions(
        simulation.Prevision(dte='2012-10-11 10:00Z', res=42, prb=50),
        simulation.Prevision(dte='2012-10-12 10:00Z', res=43, prb=50),
        simulation.Prevision(dte='2012-10-12 10:00Z', res=53, prb=33),
        simulation.Prevision(dte='2012-10-13 10:00Z', res=44, prb=100),
        simulation.Prevision(dte='2012-10-13 10:00Z', res=88, prb=50),
        simulation.Prevision(dte='2012-10-13 10:00Z', res=440, prb=0),
        simulation.Prevision(dte='2012-10-13 11:00Z', res=45, prb=0)
    )
)
print(u)
```

```
Simulation brute de H sur la Station LIMNI O034552001::<sans libelle> [0
capteur]
Date de production: <inconnue> - Qualite <inconnue>
Commentaire: <sans>
--------------------------------------------------------------------------
<modele inconnu>
--------------------------------------------------------------------------
Previsions:
 dte                  prb
2012-10-11 10:00:00  50      42
2012-10-12 10:00:00  50      43
                     33      53
2012-10-13 10:00:00  100     44
                     50      88
                     0      440
2012-10-13 11:00:00  0       45
Name: res, dtype: float64
```

### Manipuler les prévisions

In [108]:
```
# toutes les prévisions d'une probabilité donnée
print(u.previsions[:,50])
print('-' * 70)

# extraction d'une valeur
print(u.previsions['2012-10-12 10:00'])
print('-' * 70)
print(u.previsions['2012-10-12 10:00'][33]) + 1
print('-' * 70)

# slicing partiel
print(u.previsions[:'2012-10-13 10:00'])
print('-' * 70)

# slicing complet
print(u.previsions['2012-10-12 10:00Z':'2012-10-13 10:00Z'])
print('-' * 70)

# fuzzy slice
# ATTENTIOn, on ne peux pas faire un fuzzy slice directement sur le double index
print(u.previsions[:,50]['2012-10-12':'2012-10-13 23:52'])
print('-' * 70)
```

```
dte
2012-10-11 10:00:00    42
2012-10-12 10:00:00    43
2012-10-13 10:00:00    88
Name: res, dtype: float64
----------------------------------------------------------------------
prb
50     43
33     53
Name: res, dtype: float64
----------------------------------------------------------------------
54.0
----------------------------------------------------------------------
dte                  prb
2012-10-11 10:00:00  50      42
2012-10-12 10:00:00  50      43
                     33      53
2012-10-13 10:00:00  100     44
                     50      88
                     0      440
Name: res, dtype: float64
----------------------------------------------------------------------
dte                  prb
2012-10-12 10:00:00  50      43
                     33      53
2012-10-13 10:00:00  100     44
                     50      88
                     0      440
Name: res, dtype: float64
----------------------------------------------------------------------
dte
2012-10-12 10:00:00    43
2012-10-13 10:00:00    88
Name: res, dtype: float64
----------------------------------------------------------------------
```

**Une Simulation à partir du convertisseur SHOM**

In [11]:
```
from libhydro.conv import shom
import pylab

font = {'family': 'sans', 'color': 'darkgreen', 'weight': 'bold', 'size': 20}

r = shom.simulation_from_hfs('../../../test/data/shom/LOCMARIAQUER.hfs')
print(r)
r.previsions['2013-01-23 10:00':'2013-01-23 23:00'].plot(
    style='^', color='r', grid=0, linewidth=5
)
# title
pylab.title('Graphique data SHOM', fontdict=font)
pylab.text(1, 4.5, r.entite, fontdict={'family': 'sans', 'color': 'green', 'size'
# axis
pylab.xlabel('Date', fontdict=font)
pylab.ylabel('Hauteur', fontdict=font)
pylab.xticks(rotation=45)
pylab.legend(loc='upper right')
```

```
Simulation critiquee de H sur la Station LIMNI <sans code>::LOCMARIAQUER [0 capte
Date de production: <inconnue> - Qualite 100%
Commentaire: data SHOM
------------------------------------------------------------------------
Modele de type <inconnu> SCnMERshom::<sans libelle>
Description: <sans description>
------------------------------------------------------------------------
Previsions:
 dte                  prb
2013-01-23 00:00:00  50     3.46
2013-01-23 00:10:00  50     3.51
2013-01-23 00:20:00  50     3.55
2013-01-23 00:30:00  50     3.58
2013-01-23 00:40:00  50     3.62
2013-01-23 00:50:00  50     3.64
2013-01-23 01:00:00  50     3.66
2013-01-23 01:10:00  50     3.68
2013-01-23 01:20:00  50     3.70
2013-01-23 01:30:00  50     3.70
2013-01-23 01:40:00  50     3.71
2013-01-23 01:50:00  50     3.70
2013-01-23 02:00:00  50     3.70
2013-01-23 02:10:00  50     3.68
2013-01-23 02:20:00  50     3.67
...
2013-01-23 21:30:00  50     1.85
2013-01-23 21:40:00  50     1.92
2013-01-23 21:50:00  50     2.01
2013-01-23 22:00:00  50     2.09
2013-01-23 22:10:00  50     2.19
2013-01-23 22:20:00  50     2.28
2013-01-23 22:30:00  50     2.38
2013-01-23 22:40:00  50     2.48
2013-01-23 22:50:00  50     2.59
2013-01-23 23:00:00  50     2.69
2013-01-23 23:10:00  50     2.79
2013-01-23 23:20:00  50     2.89
2013-01-23 23:30:00  50     2.99
2013-01-23 23:40:00  50     3.09
2013-01-23 23:50:00  50     3.18
Name: res, Length: 144, dtype: float64
```

```
Options for the color characters are:
 'r' = red
 'g' = green
 'b' = blue
 'c' = cyan
 'm' = magenta
 'y' = yellow
 'k' = black
 'w' = white

Options for line styles are
 '-' = solid
 '--' = dashed
 ':' = dotted
 '-.' = dot-dashed
 '.' = points
 'o' = filled circles
 '^' = filled triangles
```

## Les autres modules core implémentés: intervenant et modeleprevision

In [ ]:
```
from libhydro.core import (intervenant, modeleprevision)

c = intervenant.Contact(10, 'Robert')
i = intervenant.Intervenant(code='1537', nom='SCHAPI', contacts=[c])
c.intervenant = i
print c, ' *** ', c.intervenant
```

```
Classe Contact: Classe pour manipuler des contacts.

Proprietes:
    code (entier < 9999, defaut 0)
    nom (string)
    prenom (string)
    civilite (entier parmi NOMENCLATURE[538])
    intervenant (Intervenant)


Classe Intervenant: Classe pour manipuler les intervenants Sandre.

Proprietes:
    code (int) = code SIRET (14 chiffres) ou Sandre
    origine (string in (SIRET, SANDRE)) = origine du code
    nom (string) = nom de l'intervenant
    mnemo (string) = mnemonique
    contacts (une liste de Contact)
```

In [ ]:
```
from libhydro.core import modeleprevision

m = modeleprevision.Modeleprevision('ScnMerSHOM', 'predictions du SHOM', typemode
print m
```

```
Class Modeleprevision.

Classe pour manipuler les modeles numeriques de prevision.

Proprietes:
    code (string <= 10) =
    libelle (string)
    typemodele (integer parmi NOMENCLATURE[525])
    description (string)
```

## Le convertisseur Xml Hydrométrie

**La classe Scenario:**

In [11]:
```
from libhydro.conv import xml
from libhydro.core import intervenant

emetteur = intervenant.Contact(code=45, nom='M.')
destinataire = intervenant.Intervenant(code=1537)
sce = xml.Scenario(emetteur=emetteur, destinataire=destinataire)
print '{}\n'.format(sce)
print sce.__dict__
```

```
Message du 2013-09-05 16:38:53.026417
Emis par le Contact 45::<sans civilite> M. <sans prenom> pour l'Intervenant
SANDRE 1537::<sans mnemo> [0 contact]

{'_dtprod': datetime.datetime(2013, 9, 5, 16, 38, 53, 26417),
'_destinataire': <libhydro.core.intervenant.Intervenant object at 0x2ee1dd0>,
'_emetteur': <libhydro.core.intervenant.Contact object at 0x2ee1f10>}
```

**La classe Message:**

In [13]:
```
from libhydro.conv import xml

print xml.Message.__doc__
```

```
ClasseMessage.

    Classe pour manipuler les messages Xml hydrometrie.

    Proprietes:
        scenario (xml.Scenario) = un objet Scenario obligatoire
        sitesydro (sitehydro.Sitehydro collection) = iterable ou None
        series (obshydro.Serie collection) = iterable ou None
        simulations (simulation.Simulation collection) = iterable ou None
```

**Message from xml and msg.show():**

In [23]:
```
from libhydro.conv import xml

msg = xml.Message.from_file('../../../test/data/xml/1.1/siteshydro.xml')
print '{}\n'.format(msg)
print msg.show()
```

```
Message du 2010-02-26 12:53:10
Emis par le Contact 1069::<sans civilite> <sans nom> <sans prenom> pour
l'Intervenant SANDRE 1537::<sans mnemo> [0 contact]
Contenu: 4 siteshydro - 0 series - 0 simulations

<?xml version='1.0' encoding='UTF-8'?>
<hydrometrie>
  <Scenario>
    <CodeScenario>hydrometrie</CodeScenario>
    <VersionScenario>1.1</VersionScenario>
    <NomScenario>Echange de données hydrométriques</NomScenario>
    <DateHeureCreationFichier>2010-02-26T12:53:10</DateHeureCreationFichier>
    <Emetteur>
      <CdIntervenant schemaAgencyID="SANDRE">25</CdIntervenant>
      <CdContact schemaAgencyID="SANDRE">1069</CdContact>
    </Emetteur>
    <Destinataire>
      <CdIntervenant schemaAgencyID="SANDRE">1537</CdIntervenant>
    </Destinataire>
  </Scenario>
  <RefHyd>
    <SitesHydro>
      <SiteHydro>
        <CdSiteHydro>A1984310</CdSiteHydro>
        <TypSiteHydro>REEL</TypSiteHydro>
      </SiteHydro>
      <SiteHydro>
        <CdSiteHydro>O1984310</CdSiteHydro>
        <LbSiteHydro>Le Touch à Toulouse [Saint-Martin-du-Touch]</LbSiteHydro>
        <TypSiteHydro>SOURCE</TypSiteHydro>
        <StationsHydro>
          <StationHydro>
            <CdStationHydro>O198431001</CdStationHydro>
            <LbStationHydro>station 1</LbStationHydro>
            <TypStationHydro>LIMNI</TypStationHydro>
          </StationHydro>
          <StationHydro>
            <CdStationHydro>O198431002</CdStationHydro>
            <LbStationHydro>station 2</LbStationHydro>
            <TypStationHydro>LIMNI</TypStationHydro>
          </StationHydro>
          <StationHydro>
            <CdStationHydro>O198431003</CdStationHydro>
            <LbStationHydro>station 3</LbStationHydro>
            <TypStationHydro>LIMNI</TypStationHydro>
          </StationHydro>
        </StationsHydro>
      </SiteHydro>
      <SiteHydro>
        <CdSiteHydro>O2000040</CdSiteHydro>
        <LbSiteHydro>La Garonne à Toulouse</LbSiteHydro>
        <TypSiteHydro>REEL</TypSiteHydro>
        <StationsHydro>
          <StationHydro>
            <CdStationHydro>O200004001</CdStationHydro>
```

**Ajouter des éléments à un message:**

In [16]:
```
msg2 = xml.Message.from_file('../../../test/data/xml/1.1/obsshydro.xml')
msg3 = xml.Message.from_file('../../../test/data/xml/1.1/simulations.xml')
msg.add(series=msg2.series, simulations=msg3.simulations)
print msg
```

```
Message du 2010-02-26 12:53:10
Emis par le Contact 1069::<sans civilite> <sans nom> <sans prenom> pour
l'Intervenant SANDRE 1537::<sans mnemo> [0 contact]
Contenu: 4 siteshydro - 3 series - 3 simulations
```

**Enregistrer un message dans un fichier:**

In [20]:
```
msg.write('/tmp/message.xml', force=True)
print msg.write.__doc__
```

```
Ecrit le Message dans le fichier dst.

        Cette methode est un wrapper autour de lxml.etree.ElementTree.write.
        Se referer a la documentation de lxml pour le detail des options.

        Arguments:
            dst (fichier)
            force (bool)
            encoding (string)
            compression (int de 0 a 9) = niveau de compression gzip
```