
PT3S Documentation

Release 90.12.0.2.dev1

3S

Sep 15, 2020

CONTENTS

Use SIR 3S Modeldata and SIR 3S Calculationresults in pure Python. With pandas, matplotlib and others. For documentation, test, verification, analysis, reporting, prototyping, play.

.

Use SIR 3S Modeldata and SIR 3S Results in pure Python.

With pandas, matplotlib and others.

For documentation, test, verification, analysis, reporting, prototyping, play.

SIR 3S MX-Interface (short: MX)

MX is a file based, channel-oriented interface for SIR 3S' calculation results.

This module contains stuff to utilize SIR 3S' MX calculation results in pure Python.

SIR 3S MX calculation results overview:

- Binary .MXS-Files contain the SIR 3S calculations results.
- A Model calculation run creates at least one .MXS-File.
- There is one .MX1-File (an XML-File) for the corresponding .MXS-File(s).
- This .MX1-File defines (in XML) a sequence of MX-Channels in the corresponding .MXS-File(s).
- And - as a result - the Byte-Layout of a single Record in the corresponding .MXS-File(s).
- A single Record is called a MX3-Record. .MXS-File(s) contain a sequence of MX3-Records.
- A MX3-Record contains calculation results for one TIMESTAMP. TIMESTAMP ist Scenariotime.
- A corresponding .MX3-File contains one MX3-Record - the last Scenariotime calculated.
- Summary:
- .MXS-File(s): MX3-Records: MX-Channels
- MX3-Record Byte-Layout (the MX-Channels) defined in corresponding .MX1-File.

A MX-Channel can be:

- a single Value
- or a Vector: Sequence of Values of the same Type:
 - for all Objects of a certain Type or (called Vectorchannels)
 - number of interior Points for all Pipes (special Vectorchannels: Pipevectorchannels)
 - Vectors with ATTRTYPE in: { 'SVEC', 'PVECMIN_INST', 'PVECMAX_INST' }

For Vectorchannels (including Pipevectorchannels) the sequence of Objects is defined in the .MX2-File.

```
>>> # Module Test:
>>> # -q -m 1
>>> # all Single Tests:
>>> # -q -m 0 -s Mx -s Microseconds -t both -w OneLPipe -w LocalHeatingNetwork -w_
↳ GPipe -w GPipes -w TinyWDN
>>> # ---
>>> # SETUP
```

(continues on next page)

(continued from previous page)

```

>>> # ---
>>> import os
>>> import time
>>> import logging
>>> logger = logging.getLogger('PT3S.Mx')
>>> # ---
>>> # path
>>> # ---
>>> if __name__ == "__main__":
...     try:
...         dummy=__file__
...         logger.debug("{0:s}{1:s}{2:s}".format('DOCTEST: __main__ Context: ', 'path = _
↳ os.path.dirname(__file__)', " "))
...         path = os.path.dirname(__file__)
...     except NameError:
...         logger.debug("{0:s}{1:s}{2:s}".format('DOCTEST: __main__ Context: ', "path =
↳ '.' because __file__ not defined and: ", " from Mx import Mx"))
...         path = '.'
...         from Mx import Mx
...     else:
...         logger.debug("{0:s}{1:s}{2:s}{3:s}".format('DOCTEST: Not __main__ Context: ', '_
↳ __name__: ', __name__, "path = '.'"))
...         path = '.'
>>> # ---
>>> # testDir
>>> # ---
>>> # globs={'testDir':'testdata'}
>>> try:
...     dummy= testDir
... except NameError:
...     testDir='testdata'
>>> # ---
>>> # dotResolution
>>> # ---
>>> # globs={'dotResolution':''}
>>> try:
...     dummy= dotResolution
... except NameError:
...     dotResolution=''
>>> import zipfile
>>> import pandas as pd
>>> # ---
>>> # Init
>>> # ---
>>> h5File=os.path.join(path,os.path.join(testDir,'OneLPipe.h5'))
>>> mx1File=os.path.join(path,os.path.join(testDir,'WDOneLPipe\B1\V0\BZ1\M-1-0-1
↳ '+dotResolution+'.MX1'))
>>> mx=Mx(mx1File=mx1File,NoH5Read=True,NoMxsRead=True)
>>> isinstance(mx.mx1Df,pd.core.frame.DataFrame) # MX1-Content
True
>>> isinstance(mx.df,type(None)) # MXS-Content
True
>>> # ---
>>> # Clean Up
>>> # ---
>>> mx.delFiles()
>>> if os.path.exists(mx.mxsZipFile):

```

(continues on next page)

(continued from previous page)

```

...     os.remove(mx.mxsZipFile)
>>> mxsDumpFile=mx.mxsFile+'.dump'
>>> if os.path.exists(mxsDumpFile):
...     os.remove(mxsDumpFile)
>>> # ---
>>> # 1st Read MXS
>>> # ---
>>> logger.debug("{0:s}: 1st Read MXS".format('DOCTEST'))
>>> mx.setResultsToMxsFile() # looks for M-1-0-1.MXS in same Dir
4
>>> isinstance(mx.df,pd.core.frame.DataFrame) # MXS-Content
True
>>> rowsDf,colsDf = mx.df.shape
>>> (firstTime,lastTime,rows)=mx._checkMxsVecsFile()
>>> rowsDf==rows
True
>>> mx.df.index[0]==firstTime
True
>>> # ---
>>> # Write H5
>>> # ---
>>> if os.path.exists(mx.h5File):
...     os.remove(mx.h5File)
>>> mx.ToH5() # M-1-0-1.h5 in same Dir
>>> os.path.exists(mx.h5File)
True
>>> # ---
>>> # Init with H5
>>> # ---
>>> mx=Mx(mx1File=mx1File) # looks for M-1-0-1.h5 in same Dir
>>> # and reads the .h5 if newer than .MX1 and newer than an existing .MXS
>>> isinstance(mx.mx1Df,pd.core.frame.DataFrame) # MX1-Content
True
>>> isinstance(mx.df,pd.core.frame.DataFrame) # MXS-Content
True
>>> # ---
>>> # 1st Read MXS Zip
>>> # ---
>>> # create the Zip first
>>> with zipfile.ZipFile(mx.mxsZipFile,'w') as myzip:
...     myzip.write(mx.mxsFile)
>>> logger.debug("{0:s}: 1st Read MXS Zip".format('DOCTEST'))
>>> mx.setResultsToMxsZipFile() # looks for M-1-0-1.ZIP in same Dir
4
>>> isinstance(mx.df,pd.core.frame.DataFrame) # MXS-Content
True
>>> rowsMxs,colsMxs = mx.df.shape
>>> mx.df.index.is_unique # all setResultsTo... will ensure this uniqueness under all
↳ circumstances
True
>>> # uniqueness under all circumstances: also when add=True (setResultsTo... shall
↳ add the MXS-Content) is used
>>> # ---
>>> # 1st Add same MXS (for testing ensuring uniqueness)
>>> # ---
>>> oldShape=mx.df.shape
>>> logger.debug("{0:s}: 1st Add same MXS (for testing ensuring uniqueness)".format(
↳ 'DOCTEST'))

```

(continues on next page)

(continued from previous page)

```

>>> mx.setResultsToMxsFile(add=True) # looks for M-1-0-1.MXS in same Dir
0
>>> newShape=mx.df.shape
>>> newShape==oldShape
True
>>> # ---
>>> # 1st Add same Zip (for testing ensuring uniqueness)
>>> # ---
>>> logger.debug("{0:s}: 1st Add same Zip (for testing ensuring uniqueness)".format(
↳ 'DOCTEST'))
>>> mx.setResultsToMxsZipFile(add=True) # looks for M-1-0-1.ZIP in same Dir
0
>>> newShape=mx.df.shape
>>> newShape==oldShape
True
>>> # ---
>>> # 1st Read MXS Zip with overlapping Timestamps (for testing ensuring uniqueness)
>>> # ---
>>> with zipfile.ZipFile(mx.mxsZipFile, 'w') as myzip:
...     myzip.write(mx.mxsFile)
...     myzip.write(mx.mxsFile, arcname=mx.mxsFile+'.2')
>>> logger.debug("{0:s}: 1st Read MXS Zip with overlapping Timestamps (for testing_
↳ ensuring uniqueness)".format('DOCTEST'))
>>> mx.setResultsToMxsZipFile() # looks for M-1-0-1.ZIP in same Dir
4
>>> newShape=mx.df.shape
>>> newShape==oldShape
True
>>> # ---
>>> # shift to younger Timestamps (for testing purposes)
>>> # ---
>>> lastTimestamp=mx.df.index[-1]
>>> firstTimestamp=mx.df.index[0]
>>> timeSpan=lastTimestamp-firstTimestamp
>>> if len(mx.df.index)>1:
...     timeStep=mx.df.index[-1]-mx.df.index[-2]
... else:
...     timeStep=pd.to_timedelta('1 second')
>>> mx.df.index=mx.df.index-(timeSpan+timeStep)
>>> # ---
>>> # 1st Read MXS (with the original Timestamps)
>>> # ---
>>> logger.debug("{0:s}: 1st Read MXS (with the original Timestamps)".format('DOCTEST
↳ '))
>>> mx.setResultsToMxsFile(add=True) # looks for M-1-0-1.MXS in same Dir
4
>>> rowsNew, colsNew=mx.df.shape
>>> rowsOld, colsOld=oldShape
>>> rowsNew==2*rowsOld
True
>>> colsNew==colsOld
True
>>> # ---
>>> # shift to older Timestamps (for testing purposes)
>>> # ---
>>> lastTimestamp=mx.df.index[-1]
>>> firstTimestamp=mx.df.index[0]

```

(continues on next page)

(continued from previous page)

```

>>> timeSpan=lastTimestamp-firstTimestamp
>>> if len(mx.df.index)>1:
...     timeStep=mx.df.index[-1]-mx.df.index[-2]
... else:
...     timeStep=pd.to_timedelta('1 second')
>>> mx.df.index=mx.df.index+(timeSpan+timeStep)
>>> # ---
>>> # 2nd Read MXS (with the original Timestamps)
>>> # ---
>>> logger.debug("{0:s}: 2nd Read MXS (with the original Timestamps)".format('DOCTEST
↪'))
>>> mx.setResultsToMxsFile(add=True) # looks for M-1-0-1.MXS in same Dir
4
>>> rowsNew,colsNew=mx.df.shape
>>> rowsNew==3*rowsOld
True
>>> colsNew==colsOld
True
>>> # ---
>>> # Write Dump
>>> # ---
>>> mx.dumpInMxsFormat() # dumps to .MXS.dump-File in same Dir
(12, 8)
>>> # ---
>>> # Read Dump
>>> # ---
>>> logger.debug("{0:s}: Read Dump".format('DOCTEST'))
>>> mx.setResultsToMxsFile(mxFile=mxsDumpFile)
12
>>> with zipfile.ZipFile(mx.mxsZipFile,'w') as myzip:
...     myzip.write(mx.mxsFile)
...     myzip.write(mxsDumpFile)
>>> # ---
>>> # Read Zip with Orig and Dump
>>> # ---
>>> logger.debug("{0:s}: Read Zip with Orig and Dump".format('DOCTEST'))
>>> mx.setResultsToMxsZipFile()
12
>>> (rows,cols)=mx.df.shape
>>> rows
12
>>> mx.ToH5()
>>> # ---
>>> # Without MX1, MXS
>>> # ---
>>> os.rename(mx.mx1File,mx.mx1File+'.blind')
>>> os.rename(mx.mxsFile,mx.mxsFile+'.blind')
>>> mx=Mx(mx1File=mx1File)
>>> os.rename(mx.mx1File+'.blind',mx.mx1File)
>>> os.rename(mx.mxsFile+'.blind',mx.mxsFile)
>>> # ---
>>> sir3sIdTimestamp=mx.mx1Df['Sir3sID'].iloc[mx.idxTIMESTAMP]
>>> mx.mx1Df['Sir3sID'][mx.mx1Df['Sir3sID']==sir3sIdTimestamp].index[0]
0
>>> mx.mx1Df['unpackIdx'][mx.mx1Df['Sir3sID']==sir3sIdTimestamp].iloc[0]
0
>>> (rows,cols)=mx.df.shape

```

(continues on next page)

(continued from previous page)

```

>>> rows
12
>>> isinstance(mx.df.index[0],pd.Timestamp)
True
>>> str(mx.df.index[0])
'2018-03-03 00:00:00+00:00'
>>> sir3sId=mx.mx1Df['Sir3sID'][mx.mx1Df['Sir3sID'].str.contains('KNOT~\S*~
↳5642914844465475844~QM')].iloc[0] #KNOT~I~~5642914844465475844~QM
>>> ts=mx.df[sir3sId]
>>> isinstance(ts,pd.core.series.Series)
True
>>> "{:06.2f}".format(round(ts.iloc[0],2))
'176.71'
>>> logger.debug("{:s}: CHANGEHISTORY: {:>10s}: {:>3d}: {:>6s}: {:s}".format('DOCTEST
↳', '0.0.31',1,'New','_checkMxsVecsFile: (... ,fullCheck=False,...)'))
>>> mx._checkMxsVecsFile()
(Timestamp('2018-03-03 00:00:00+0000', tz='UTC'), Timestamp('2018-03-03 00:00:11+0000
↳', tz='UTC'), 12)
>>> mx._checkMxsVecsFile(fullCheck=True)
(Timestamp('2018-03-03 00:00:00+0000', tz='UTC'), Timestamp('2018-03-03 00:00:11+0000
↳', tz='UTC'), 12)
>>> logger.debug("{:s}: CHANGEHISTORY: {:>10s}: {:>3d}: {:>6s}: {:s}".format('DOCTEST
↳', '0.0.32',1,'Change','setResultsToMxsFile: finally: h5.close()'))
>>> logger.debug("{:s}: CHANGEHISTORY: {:>10s}: {:>3d}: {:>6s}: {:s}".format('DOCTEST
↳', '0.0.32',2,'Change','setResultsToMxsZipFile: finally: h5.close()'))
>>> logger.debug("{:s}: CHANGEHISTORY: {:>10s}: {:>3d}: {:>6s}: {:s}".format('DOCTEST
↳', '0.0.32',3,'Change','ToH5: finally: h5.close()'))
>>> logger.debug("{:s}: CHANGEHISTORY: {:>10s}: {:>3d}: {:>6s}: {:s}".format('DOCTEST
↳', '0.0.32',4,'Change','FromH5: finally: h5.close()'))
>>> logger.debug("{:s}: CHANGEHISTORY: {:>10s}: {:>3d}: {:>6s}: {:s}".format('DOCTEST
↳', '0.0.32',5,'Change','*: except Exception as e'))
>>> logger.debug("{:s}: CHANGEHISTORY: {:>10s}: {:>3d}: {:>6s}: {:s}".format('DOCTEST
↳', '0.0.32',6,'Change','setResultsToMxsFile: finally: NewH5Vec=False'))
>>> logger.debug("{:s}: CHANGEHISTORY: {:>10s}: {:>3d}: {:>6s}: {:s}".format('DOCTEST
↳', '0.0.32',7,'Change','__init__(...,NoH5Read=True,...)'))
>>> if os.path.exists(mx.h5FileVecs):
...     os.remove(mx.h5FileVecs)
>>> mx=Mx(mx1File=mx1File,NoH5Read=True)
>>> os.path.exists(mx.h5FileVecs) # h5 was written (h5 did not exist)
True
>>> h5VecsFileTime=os.path.getmtime(mx.h5FileVecs)
>>> mx=Mx(mx1File=mx1File,NoH5Read=True) # h5 will be written again (h5 exists)
>>> h5VecsFileTimeNow=os.path.getmtime(mx.h5FileVecs)
>>> logger.debug("1 h5VecsFileTime:{:s} < h5VecsFileTimeNow:{:s}".format(time.
↳strftime("%Y-%m-%d %H:%M:%S %z",time.gmtime(h5VecsFileTime)),time.strftime("%Y-%m-
↳%d %H:%M:%S %z",time.gmtime(h5VecsFileTimeNow))))
>>> h5VecsFileTime<h5VecsFileTimeNow # 1 h5 was written again (h5 did exist before)
True
>>> h5VecsFileTime=os.path.getmtime(mx.h5FileVecs)
>>> mx=Mx(mx1File=mx1File) # h5 is read - not written
>>> h5VecsFileTimeNow=os.path.getmtime(mx.h5FileVecs)
>>> logger.debug("2 h5VecsFileTime:{:s} == h5VecsFileTimeNow:{:s}".format(time.
↳strftime("%Y-%m-%d %H:%M:%S %z",time.gmtime(h5VecsFileTime)),time.strftime("%Y-%m-
↳%d %H:%M:%S %z",time.gmtime(h5VecsFileTimeNow))))
>>> mx.setResultsToMxsFile() # h5 will not be updated
0
>>> h5VecsFileTimeNow=os.path.getmtime(mx.h5FileVecs)

```

(continues on next page)

(continued from previous page)

```

>>> logger.debug("3 h5VecsFileTime:{:s} == h5VecsFileTimeNow:{:s}".format(time.
↳ strftime("%Y-%m-%d %H:%M:%S %z",time.gmtime(h5VecsFileTime)),time.strftime("%Y-%m-
↳ %d %H:%M:%S %z",time.gmtime(h5VecsFileTimeNow))))
>>> mx.setResultsToMxsFile(NewH5Vec=True) # h5 is written
4
>>> h5VecsFileTimeNow=os.path.getmtime(mx.h5FileVecs)
>>> logger.debug("h5VecsFileTime:{:s} == h5VecsFileTimeNow:{:s}".format(time.strftime(
↳ "%Y-%m-%d %H:%M:%S %z",time.gmtime(h5VecsFileTime)),time.strftime("%Y-%m-%d %H:%M:
↳ %S %z",time.gmtime(h5VecsFileTimeNow))))
>>> logger.debug("{:s}: CHANGEHISTORY: {:>10s}: {:>3d}: {:>6s}: {:s}".format('DOCTEST
↳ ', '0.0.32',8,'Change','setResultsToMxsZipFile: finally: NewH5Vec=False'))
>>> mx.ToH5()
>>> mx=Mx(mx1File=mx1File,NoH5Read=True)
>>> os.path.exists(mx.h5File)
False
>>> #pd.set_option('display.max_columns',None)
>>> #pd.set_option('display.max_rows',None)
>>> #pd.set_option('display.max_colwidth',666666)
>>> #pd.set_option('display.width',666666666)
>>> print(mx._getMx1DfAsOneString().replace('\n','\n '))
ATTRTYPE  DATALENGTH  DATATYPE  DATATYPELENGTH  FLAGS  OBJTYPE  OBJTYPE_PK
↳
↳      TITLE  UNIT
↳      PH      4      REAL      4      1265      KNOT      5289899964753656852
↳      Druck [bar]
↳      QM      4      REAL      4      1265      KNOT      5289899964753656852
↳ Externer Durchfluss [m3/h]
↳      PH      4      REAL      4      1265      KNOT      5642914844465475844
↳      Druck [bar]
↳      QM      4      REAL      4      1265      KNOT      5642914844465475844
↳ Externer Durchfluss [m3/h]
>>> print("'{:s}'".format(repr(mx.mx2Df).replace('\n','\n ')))
'''      ObjType      AttrType  DataType  DataTypeLength  DataLength  NOItems
↳
↳      Data
↳      0  KNOT      tk      CHAR      20      40      2
↳ [5642914844465475844, 5289899964753656852]
↳      1  LFKT      pk      CHAR      20      20      1
↳ [5252810657060947333]
↳      2  PHI1      pk      CHAR      20      20      1
↳ [5502689500012692689]
↳      3  PUMD      pk      CHAR      20      20      1
↳ [5732781659713982525]
↳      4  PVAR      pk      CHAR      20      20      1
↳ [5163733225086798083]
↳      5  QVAR      pk      CHAR      20      20      1
↳ [4742976321174242828]
↳      6  ROHR      tk      CHAR      20      20      1
↳ [4737064599036143765]
↳      7  ROHR      N_OF_POINTS  INT4      4      4      1
↳ (101,)
↳      8  SWVT      pk      CHAR      20      20      1
↳ [5396761270498593493]'''
>>> print(mx._getDfAsOneString())
KNOT~~~5289899964753656852~PH KNOT~~~5289899964753656852~QM
↳ KNOT~~~5642914844465475844~PH KNOT~~~5642914844465475844~QM
2018-03-03 00:00:00+00:00      0.0      -176.7
↳      4.2      176.7
2018-03-03 00:00:01+00:00      0.0      -176.7
↳      4.2      176.7

```

(continues on next page)

(continued from previous page)

```

2018-03-03 00:00:02+00:00          0.0          -176.7
↪          4.2          176.7
2018-03-03 00:00:03+00:00          0.0          -176.7
↪          4.2          176.7
>>> # ---
>>> # Clean Up OneLPipe
>>> # ---
>>> mx.delFiles()
>>> if os.path.exists(mx.mxsZipFile):
...     os.remove(mx.mxsZipFile)
>>> mxsDumpFile=mx.mxsFile+'.dump'
>>> if os.path.exists(mxsDumpFile):
...     os.remove(mxsDumpFile)
>>> # ---
>>> # LocalHeatingNetwork
>>> # ---
>>> mx1File=os.path.join(path,os.path.join(testDir,'WDLocalHeatingNetwork\B1\V0\BZ1\M-
↪1-0-1'+dotResolution+'.MX1'))
>>> mx=Mx(mx1File=mx1File,NoH5Read=True,NoMxsRead=True)
>>> # ---
>>> # Clean Up LocalHeatingNetwork
>>> # ---
>>> mx.delFiles()
>>> if os.path.exists(mx.mxsZipFile):
...     os.remove(mx.mxsZipFile)
>>> mxsDumpFile=mx.mxsFile+'.dump'
>>> if os.path.exists(mxsDumpFile):
...     os.remove(mxsDumpFile)
>>> mx.setResultsToMxsFile(maxRecords=1)
1
>>> print(mx._getDfAsOneString())
          KNOT~~~5356267303828212700~PH KNOT~~~5397990465339071638~QM
↪KNOT~~~5736262931552588702~PH
2004-09-22 08:30:00+00:00          2.0          0.0
↪          4.1
>>> logger.debug("{:s}: CHANGEHISTORY: {:>10s}: {:>3d}: {:>6s}: {:s}".format('DOCTEST
↪','0.0.41',1,'New',"getMxsVecsFileData"))
>>> timesReq=[]
>>> timesReq.append(mx.df.index[0])
>>> plotTimeDfs=mx.getMxsVecsFileData(timesReq=timesReq)
>>> len(plotTimeDfs)
1
>>> isinstance(plotTimeDfs[0],pd.core.frame.DataFrame)
True
>>> print(mx._getDfVecAsOneString(df=plotTimeDfs[0],regex='KNOT~\S*~\S*~\S*~T$'))
          KNOT~~~~T
2004-09-22 08:30:00+00:00          60.00
>>> timesReq[0]=timesReq[0]-pd.to_timedelta('1 second')
>>> plotTimeDfs=mx.getMxsVecsFileData(timesReq=timesReq)
>>> len(plotTimeDfs)
0
>>> # ---
>>> # Clean Up LocalHeatingNetwork
>>> # ---
>>> mx.delFiles()
>>> if os.path.exists(mx.mxsZipFile):
...     os.remove(mx.mxsZipFile)

```

(continues on next page)

(continues on next page)

(continued from previous page)

```

5   PVAR                pk                CHAR                20                20                1
[4728046487162557851]
6   QVAR                pk                CHAR                20                20                1
[5377391575499483345]
7   ROHR                tk                CHAR                20                440                22
[4820249423461369400, 5508607066407516090, 5095827458323717990, 5700891034663173564,
5729543164571807746, 5067763789937102564, 4799443678518131207, 5605831797210762998,
5126294008398890918, 5419686039699182683, 5312551747276023841, 5296152509037292880,
5670035893444309530, 5433880705192526755, 4648047345314768819, 4984438795139137900,
5644872080928983958, 5148090523913666712, 5461179577260327606, 4978978527327130204,
5497762617222653432, 5076321356874807093]
8   ROHR                N_OF_POINTS      INT4                4                88                22
(5, 6, 5, 3, 4, 3, 3, 3, 3,
3, 3, 3, 3, 2, 5, 4, 3, 3, 4, 3, 2, 2)
9   SWVT                pk                CHAR                20                20                1
[4955975670134047353]
10  TEVT                pk                CHAR                20                20                1
[4821755615165519990]
11  VENT                tk                CHAR                20                20                1
[5466655470152247657]
12  WEVT                pk                CHAR                20                20                1
[5736734929574151957]'''
>>> mx.setResultsToMxsFile(maxRecords=1)

```

(continues on next page)

(continued from previous page)

```

1
>>> print(mx._getDfAsOneString())
                                KNOT~~~4711309381204507891~PH KNOT~~~5179406559406617933~PH
↪KNOT~~~5179406559406617933~QM
2002-05-22 16:16:16+00:00          1.0          3.0
↪          140.0
>>> # ---
>>> # Clean Up Tiny WDN
>>> # ---
>>> mx.delFiles()
>>> # ---
>>> # GPipe
>>> # ---
>>> mx1File=os.path.join(path,os.path.join(testDir,'WDGPipe\B1\V0\BZ1\M-1-0-1
↪'+dotResolution+'.MX1'))
>>> mx=Mx(mx1File=mx1File)
>>> plotTimeDfs=mx.getMxsVecsFileData()
>>> len(plotTimeDfs[0]['ROHR~*~*~*~PVEC'].iloc[0]) == mx.mx2Df[mx.mx2Df['AttrType']
↪str.contains('N_OF_POINTS')].iloc[0].Data[0]
True
>>> plotTimeDfs[0]['ROHR~*~*~*~PVEC'].iloc[0][0]
41.0
>>> # ---
>>> # Clean Up GPipe
>>> # ---
>>> mx.delFiles()

```

class `Mx.Mx` (*mx1File*, *NoH5Read=False*, *NoMxsRead=False*, *maxRecords=None*)
 Reading SIR 3S' MX-Files.

Summary:

- (*mx1File*): use this to profit from previous reads finalized with `ToH5()`
- (*mx1File*, *NoH5Read=True*): use this for a fresh start with implicit `.MXS`-File read; finalize with `ToH5()`
- (*mx1File*, *NoH5Read=True*, *NoMxsRead=True*): use this for a fresh start; call `setResultsTo...`() explicit; finalize with `ToH5()`
- note that `base.Y.h5`-File has to be dunned explicitly with `ToH5()`
- and `base.Y.vec.h5`-File is written implicitly while (implicit or explicit calls to) `setResultsTo...`()
- and is deleted explicitly (*mx1File*, *NoH5Read=True*) or implicitly (because it is i.e. to old)

Args:

- *mx1File* (str): `base.MX1`-File (an XML-File) (`base.Y.MX1`-File from 90-10 on)
- **NoH5Read** (bool):

False (default - use this to profit from previous reads finalized with `ToH5()`):

– If a `base.Y.h5`-File

- * exists
- * and is newer (>) than an `.MX1`-File (`base.Y.Mx1`-File from 90-10 on)
- * and is newer (>) than an `.MXS`-File (`base.Y.MXS`-File from 90-10 on):
 - The `base.Y.h5`-File is read instead of the `.MX1`-File.

True (use this for a fresh start):

- An base.Y.h5-File is deleted if existing.
- The base.Y.Mx1-File is read.
- The base.Y.vec.h5-File is newly created in case of an .MXS-File read.

• NoMxsRead (bool):**True:**

- a base.Y.MXS-File is not read
- a base.Y.vec.h5-File is not touched

False (default):**– If a base.Y.MXS-File**

- * exists
- * and is newer (\geq) than base.Y.Mx1-File
- * and base.Y.h5-File is not read:
 - The base.Y.MXS-File is read.
 - NoH5Read=True will delete base.Y.vec.h5-File.

• maxRecords (default: None):

- number of records to be read from base.Y.MXS-File
- by default all records are read
- use maxRecords only for Test purposes

Attributes:**• states**

- h5Read: True, if read from H5

• fileNames

- .mx1File: base.Y.MX1-File

– derived from mx1File

- * .mx2File: base.MX2-File
- * .mxsFile: base.Y.MXS-File
- * .mxsZipFile base.ZIP
- * constructed from MX during Init and Usage:
- *
- * .h5File: base.Y.h5-File
- * .h5FileVecs: base.Y.vec.h5-File: MXS-H5Dump written implicitly
- * .h5FileMx1FmtString: base.Y.h5-File.metadata written implicitly

• .mxRecordStructFmtString

- usage: struct.unpack(self.mxRecordStructFmtString,a_MXS_Record)
- .h5FileMx1FmtString:

- * it was not possible to store mxRecordStructFmtString in H5-Format as Metadata
- * therefore mxRecordStructFmtString is stored in a file named .h5FileMx1FmtString
- * in .h5File the Link to this file is stored as Metadata
- * as pointed out with usage above mxRecordStructFmtString has nothing to do with writing to or reading from H5
- * mxRecordStructFmtString is only about reading from (and writing to for test purposes) MXS
- * the .mxRecordStructFmtString/.h5FileMx1FmtString stuff is only about performance:
- * if after reading from H5 only a(nother) MXS shall be read again ...
- * ... the stuff avoids the time-consuming reconstruction of mxRecordStructFmtString

- **dataFrames**

- .mx1Df
- .mx2Df
- **.df**
 - * the base.Y.MXS-File(s) Content
 - * non Vectordata only
 - * index: TIMESTAMP (scenario time)
 - * **columns: Values**
 - The following (String-)ID - called Sir3sID - is used as Columnlabel:
 - this Sir3sID consists of ~ (Mx.reSir3sIDSep) separated .MX1-File terms:
 - OBJTYPE~NAME1~NAME2~OBJTYPE_PK~ATTRTYPE
 - Sir3sID regExp Example: Mx.reSir3sID: (?P<OBJTYPE>S+)(~(?P<NAME1>[S]*))~(?P<NAME1>S*)~(?P<OBJTYPE_PK>d+)~(?P<ATTRTYPE>S+)
- **.dfVecAggs**
 - * some base.Y.vec.h5-File (Vectordata only) Aggregates as df
 - * **MultiIndex:**
 - TYPE: SNAPSHOTTYPES: TIME,TMIN,TMAX (from _readMxsFile) or Aggregates from 2 Times: MIN,MAX,... (from getVecAggs)
 - Sir3sID
 - TIMESTAMPL
 - TIMESTAMPR
 - * Cols: mx2Idx
 - * Value: Aggregate (i.e. TMIN) for mx2Idx-Col

Raises: MxError

```
>>> mx=mxs [ 'LocalHeatingNetwork' ]
>>> try:
...     import Mx
... except:
```

(continues on next page)

(continued from previous page)

```

...     from PT3S import Mx
>>> mx=Mx.Mx(mx.mx1File)
>>> mx.df.filter(items=['ALLG~~~4639827058859487185~SNAPSHOTTYPE', 'KNOT~V-L~~
↳5736262931552588702~PH']).round(1)
      ALLG~~~4639827058859487185~SNAPSHOTTYPE  KNOT~V-L~~
↳5736262931552588702~PH
2004-09-22 08:30:00+00:00      b'STAT'
↳      4.1
2004-09-22 08:30:15+00:00      b'TIME'
↳      3.2
2004-09-22 08:30:30+00:00      b'TIME'
↳      2.6
2004-09-22 08:30:45+00:00      b'TIME'
↳      2.2
2004-09-22 08:31:00+00:00      b'TIME'
↳      4.1
>>> mx=Mx.Mx(mx.mx1File,NoH5Read=True,maxRecords=1)
>>> mx.df.filter(items=['ALLG~~~4639827058859487185~SNAPSHOTTYPE', 'KNOT~V-L~~
↳5736262931552588702~PH']).round(1)
      ALLG~~~4639827058859487185~SNAPSHOTTYPE  KNOT~V-L~~
↳5736262931552588702~PH
2004-09-22 08:30:00+00:00      b'STAT'
↳      4.1
>>> mx=Mx.Mx(mx.mx1File,NoH5Read=True,maxRecords=2)
>>> mx.df.filter(items=['ALLG~~~4639827058859487185~SNAPSHOTTYPE', 'KNOT~V-L~~
↳5736262931552588702~PH']).round(1)
      ALLG~~~4639827058859487185~SNAPSHOTTYPE  KNOT~V-L~~
↳5736262931552588702~PH
2004-09-22 08:30:00+00:00      b'STAT'
↳      4.1
>>> mx=Mx.Mx(mx.mx1File,NoH5Read=True,maxRecords=3)
>>> mx.df.filter(items=['ALLG~~~4639827058859487185~SNAPSHOTTYPE', 'KNOT~V-L~~
↳5736262931552588702~PH']).round(1)
      ALLG~~~4639827058859487185~SNAPSHOTTYPE  KNOT~V-L~~
↳5736262931552588702~PH
2004-09-22 08:30:00+00:00      b'STAT'
↳      4.1
2004-09-22 08:30:15+00:00      b'TIME'
↳      3.2

```

delFiles()

Deletes Files constructed by MX during Init and Usage.

_initWithMx1()

(Re-)initialize .mx1Df, .mx2Df, .mxRecordStructFmtString and related stuff with .mx1File.

Calls:

- ._parseMx1()
- ._parseMx2()
- ._buildMxRecordStructUnpackFmtString()
- ._buildMxRecordStructUnpackFmtStringPost()

Raises: MxError**_parseMx1()**

Parses .mx1File.

Sets

- .mx1Df

Raises: MxError

_parseMx2 ()

Parses .mx2File.

Sets

- .mx2Df

Raises: MxError

_buildMxRecordStructUnpackFmtString ()

(Re-)builds .mxRecordStructFmtString and related stuff.

Sets

- .mxRecordStructFmtString
- .mx1Df['unpackIdx']

Raises: MxError

_buildMxRecordStructUnpackFmtStringPost ()

Stuff todo after buildMxRecordStructUnpackFmtString.

Sets

- .idxCVERSO (idx of CVERSO in MX1)
- .unpackIdxCVERSO (idx of CVERSO in recordData)
- .idxTIMESTAMP (idx of TIMESTAMP in MX1)
- .unpackIdxTIMESTAMP (idx of TIMESTAMP in recordData)
- .mxColumnNames=[] (of non Vectordata without TIMESTAMP in MX1-Sequence)
- .mxColumnNamesVecs=[] (of Vectordata without TIMESTAMP in MX1-Sequence)
- .idxUnpackNonVectorChannels[] (idx in recordData)
- .idxUnpackVectorChannels[] (idx in recordData of the 1st ([0]) Element of the Vector)
- .idxOfNonVectorChannels[] (idx in MX1 without TIMESTAMP)
- .idxVectorChannels[] (idx in MX1)

Raises: MxError

_readMxsFile (mxsFilePtr, mxsVecsH5StorePtr, firstTime=None, maxRecords=None)**Args:**

- mxsFilePtr: .MXS-File
- mxsVecsH5StorePtr: .vec.h5-File (Vectordata only)
- **firstTime: used to calculate h5Key for mxsVecsH5Store**
 - None (default): firstTime is set to 1st TIMESTAMP in .MXS-File
 - **else:**
 - * caller sets firstTime - in general the youngest TIMESTAMP in the df
- maxRecords: Anzahl der max. zu lesenden Records

(continued from previous page)

```

TIME KNOT~~~~~PH 2004-09-22 08:30:00 2004-09-22 08:30:00 2.3 4.0 4.1 4.
↪1 2.0 2.3 2.0 4.3 4.1 2.3 4.3 2.0 2.1 3.8 2.0 3.8 2.3 3.8 2.
↪3 2.0 3.8 4.1 3.8
TMIN KNOT~~~~~PH 2004-09-22 08:30:00 2004-09-22 08:31:00 2.1 2.2 2.2 2.
↪2 2.0 2.0 2.0 2.2 2.2 2.1 2.2 2.0 2.0 2.2 2.0 2.2 2.1 2.2 2.
↪1 2.0 2.2 2.2 2.2
TMAX KNOT~~~~~PH 2004-09-22 08:30:00 2004-09-22 08:31:00 2.3 4.0 4.1 4.
↪1 2.0 2.3 2.0 4.3 4.1 2.3 4.3 2.0 2.1 3.8 2.0 3.8 2.3 3.8 2.
↪3 2.0 3.8 4.1 3.8
>>> mx.dfVecAggs.round(1)
                                0
↪ 1      2      3      4      5      6      7      8      9      10     11
↪ 12     13     14     15     16     17     18     19     20     21     22
↪ 23     24     25     26     27     28     29     30     31
TYPE Sir3sID                    TIMESTAMPL                    TIMESTAMPR
↪
↪
↪
TIME Klap~~~~~QM                    2004-09-22 08:30:00 2004-09-22 08:30:00 23.0
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
ROHR~~~~~RHOVEC                    2004-09-22 08:30:00 2004-09-22 08:30:00 983.7
↪ 983.7 965.7 965.7 983.7 983.7 965.7 965.7 983.7 983.7 965.7 965.7
↪ 965.7 965.7 983.7 983.7 965.7 965.7 965.7 965.7 965.7 965.7 983.
↪ 7 983.7 983.7 983.7 983.7 983.7 965.7 965.7 983.7 983.7
ROHR~~~~~TVEC                    2004-09-22 08:30:00 2004-09-22 08:30:00 60.0
↪ 60.0 90.0 90.0 60.0 60.0 90.0 90.0 60.0 60.0 90.0 90.0
↪ 90.0 90.0 60.0 60.0 90.0 90.0 90.0 90.0 90.0 90.0 60.0
↪ 60.0 60.0 60.0 60.0 60.0 90.0 90.0 60.0 60.0
FWES~~~~~V                    2004-09-22 08:30:00 2004-09-22 08:30:00 1.3
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
ROHR~~~~~VAV                    2004-09-22 08:30:00 2004-09-22 08:30:00 -0.3
↪ 0.6 -0.5 0.3 -0.3 0.7 0.3 -0.1 0.7 0.5 0.1 -0.7
↪ -0.6 -0.1 0.1 -0.7 NaN NaN NaN NaN NaN NaN
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
KNOT~~~~~IAKTIV                    2004-09-22 08:30:00 2004-09-22 08:30:00 0.0
↪ 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
↪ 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
ROHR~~~~~IAKTIV                    2004-09-22 08:30:00 2004-09-22 08:30:00 0.0
↪ 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
↪ 0.0 0.0 0.0 0.0 0.0 NaN NaN NaN NaN NaN NaN
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
ROHR~~~~~MVEC                    2004-09-22 08:30:00 2004-09-22 08:30:00 -2.4
↪ -2.4 5.3 5.3 -4.3 -4.3 2.4 2.4 -6.4 -6.4 6.4 6.4
↪ 6.4 6.4 -1.1 -1.1 6.4 6.4 4.3 4.3 1.1 1.1 -6.4
↪ -6.4 -5.3 -5.3 -1.1 -1.1 1.1 1.1 -6.4 -6.4
ROHR~~~~~VK                    2004-09-22 08:30:00 2004-09-22 08:30:00 -0.3
↪ 0.6 -0.5 0.3 -0.3 0.7 0.3 -0.1 0.7 0.5 0.1 -0.7
↪ -0.6 -0.1 0.1 -0.7 NaN NaN NaN NaN NaN NaN
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
PUMP~~~~~QM                    2004-09-22 08:30:00 2004-09-22 08:30:00 23.0
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
↪ NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN

```

(continues on next page)

(continued from previous page)

	KNOT~~~~~PDAMPF			2004-09-22	08:30:00	2004-09-22	08:30:00	0.2			
→	0.7	0.7	0.7	0.2	0.2	0.2	0.2	0.7	0.2	0.2	0.2
→	0.2	0.7	0.2	0.7	0.2	0.7	0.2	0.2	0.7	0.7	0.7
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	ROHR~~~~~QMI			2004-09-22	08:30:00	2004-09-22	08:30:00	-8.5			
→	19.1	-15.4	8.5	-23.0	23.0	23.0	-3.9	23.0	15.4	3.9	-23.0
→	-19.1	-3.9	3.9	-23.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	KNOT~~~~~WALTER			2004-09-22	08:30:00	2004-09-22	08:30:00	0.2			
→	0.2	0.1	0.1	0.4	0.2	0.4	0.5	0.0	0.3	0.5	0.5
→	0.3	0.4	0.0	0.5	0.0	0.8	0.2	0.5	0.3	0.0	1.1
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	KNOT~~~~~P			2004-09-22	08:30:00	2004-09-22	08:30:00	3.3			
→	5.0	5.1	5.1	3.0	3.3	3.0	5.3	5.1	3.3	5.3	3.0
→	3.1	4.8	3.0	4.8	3.3	4.8	3.3	3.0	4.8	5.1	4.8
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	FWES~~~~~QM			2004-09-22	08:30:00	2004-09-22	08:30:00	23.0			
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	KNOT~~~~~HMAX_INST			2004-09-22	08:30:00	2004-09-22	08:30:00	2.3			
→	4.0	4.1	4.1	2.0	2.3	2.0	4.3	4.1	2.3	4.3	2.0
→	2.1	3.8	2.0	3.8	2.3	3.8	2.3	2.0	3.8	4.1	3.8
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	ROHR~~~~~PVECMIN_INST			2004-09-22	08:30:00	2004-09-22	08:30:00	3.3			
→	3.3	5.0	4.8	3.3	3.3	4.8	4.8	3.0	3.0	5.1	5.0
→	5.1	5.1	3.3	3.3	5.1	5.1	4.8	4.8	4.8	4.8	3.0
→	3.1	3.1	3.3	3.3	3.3	4.8	4.8	3.0	3.0		
	VENT~~~~~QM			2004-09-22	08:30:00	2004-09-22	08:30:00	23.0			
→	23.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	KNOT~~~~~PH			2004-09-22	08:30:00	2004-09-22	08:30:00	2.3			
→	4.0	4.1	4.1	2.0	2.3	2.0	4.3	4.1	2.3	4.3	2.0
→	2.1	3.8	2.0	3.8	2.3	3.8	2.3	2.0	3.8	4.1	3.8
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	FWVB~~~~~W			2004-09-22	08:30:00	2004-09-22	08:30:00	160.0			
→	200.0	160.0	160.0	120.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	FWVB~~~~~QM			2004-09-22	08:30:00	2004-09-22	08:30:00	3.9			
→	6.9	4.6	3.9	3.7	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	KNOT~~~~~H			2004-09-22	08:30:00	2004-09-22	08:30:00	2.3			
→	4.0	4.1	4.1	2.0	2.3	2.0	4.3	4.1	2.3	4.3	2.0
→	2.1	3.8	2.0	3.8	2.3	3.8	2.3	2.0	3.8	4.1	3.8
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	ROHR~~~~~SVEC			2004-09-22	08:30:00	2004-09-22	08:30:00	0.0			
→	88.0	0.0	406.0	0.0	83.6	0.0	88.0	0.0	73.4	0.0	195.5
→	0.0	68.6	0.0	109.8	0.0	76.4	0.0	83.6	0.0	164.9	0.0
→	195.5	0.0	406.0	0.0	164.9	0.0	109.8	0.0	76.4		
	ROHR~~~~~PVEC			2004-09-22	08:30:00	2004-09-22	08:30:00	3.3			
→	3.3	5.0	4.8	3.3	3.3	4.8	4.8	3.0	3.0	5.1	5.0
→	5.1	5.1	3.3	3.3	5.1	5.1	4.8	4.8	4.8	4.8	3.0
→	3.1	3.1	3.3	3.3	3.3	4.8	4.8	3.0	3.0		
	ROHR~~~~~QMAV			2004-09-22	08:30:00	2004-09-22	08:30:00	-8.5			
→	19.1	-15.4	8.5	-23.0	23.0	23.0	-3.9	23.0	15.4		
→	-19.1	-3.9	3.9	-23.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		

(continues on next page)

(continued from previous page)

	ROHR~~~~~VI	2004-09-22 08:30:00	2004-09-22 08:30:00	-0.3	
→	0.6 -0.5 0.3 -0.3 0.7 0.3 -0.1 0.7 0.5 0.1 -0.7				
→	-0.6 -0.1 0.1 -0.7 NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	KNOT~~~~~T	2004-09-22 08:30:00	2004-09-22 08:30:00	60.0	
→	90.0 90.0 90.0 60.0 60.0 60.0 60.0 90.0 60.0 60.0 60.0				
→	60.0 90.0 60.0 90.0 60.0 90.0 60.0 60.0 90.0 90.0 90.0				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	VENT~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:30:00	0.0	
→	0.0 0.0 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	ROHR~~~~~PVECMAX_INST	2004-09-22 08:30:00	2004-09-22 08:30:00	3.3	
→	3.3 5.0 4.8 3.3 3.3 4.8 4.8 3.0 3.0 5.1 5.0				
→	5.1 5.1 3.3 3.3 5.1 5.1 4.8 4.8 4.8 4.8 3.0				
→	3.1 3.1 3.3 3.3 3.3 4.8 4.8 3.0 3.0				
	VENT~~~~~V	2004-09-22 08:30:00	2004-09-22 08:30:00	0.4	
→	0.4 0.0 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	KNOT~~~~~HMIN_INST	2004-09-22 08:30:00	2004-09-22 08:30:00	2.3	
→	4.0 4.1 4.1 2.0 2.3 2.0 4.3 4.1 2.3 4.3 2.0				
→	2.1 3.8 2.0 3.8 2.3 3.8 2.3 2.0 3.8 4.1 3.8				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	ROHR~~~~~QMK	2004-09-22 08:30:00	2004-09-22 08:30:00	-8.5	
→	19.1 -15.4 8.5 -23.0 23.0 23.0 -3.9 23.0 15.4 3.9 -23.0				
→	-19.1 -3.9 3.9 -23.0 NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	KNOT~~~~~PMIN_INST	2004-09-22 08:30:00	2004-09-22 08:30:00	3.3	
→	5.0 5.1 5.1 3.0 3.3 3.0 5.3 5.1 3.3 5.3 3.0				
→	3.1 4.8 3.0 4.8 3.3 4.8 3.3 3.0 4.8 5.1 4.8				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	KNOT~~~~~RHO	2004-09-22 08:30:00	2004-09-22 08:30:00	983.7	
→	965.7 965.7 965.7 983.7 983.7 983.7 965.7 983.7 983.7 983.7				
→	983.7 965.7 983.7 965.7 983.7 965.7 983.7 983.7 965.7 965.				
→	7 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	ROHR~~~~~ZVEC	2004-09-22 08:30:00	2004-09-22 08:30:00	20.0	
→	20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0				
→	20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0				
→	20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0				
	KNOT~~~~~PMAX_INST	2004-09-22 08:30:00	2004-09-22 08:30:00	3.3	
→	5.0 5.1 5.1 3.0 3.3 3.0 5.3 5.1 3.3 5.3 3.0				
→	3.1 4.8 3.0 4.8 3.3 4.8 3.3 3.0 4.8 5.1 4.8				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	KLAP~~~~~V	2004-09-22 08:30:00	2004-09-22 08:30:00	1.3	
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	KLAP~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:30:00	0.0	
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	PUMP~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:30:00	0.0	
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	FWVB~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:30:00	0.0	
→	0.0 0.0 0.0 0.0 NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				

(continues on next page)

(continued from previous page)

	FWES~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:30:00	0.0	
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	TMIN KLAP~~~~~QM	2004-09-22 08:30:00	2004-09-22 08:31:00	9.2	
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	ROHR~~~~~RHOVEC	2004-09-22 08:30:00	2004-09-22 08:31:00	983.7	
→	983.7 965.7 965.7 983.7 983.7 965.7 965.7 983.7 983.7 965.7 965.7				
→	965.7 965.7 983.7 983.7 965.7 965.7 965.7 965.7 965.7 965.7 983.7				
→	7 983.7 983.7 983.7 983.7 983.7 965.7 965.7 983.7 983.7				
	ROHR~~~~~TVEC	2004-09-22 08:30:00	2004-09-22 08:31:00	60.0	
→	60.0 90.0 90.0 60.0 60.0 90.0 90.0 60.0 60.0 90.0 90.0				
→	90.0 90.0 60.0 60.0 90.0 90.0 90.0 90.0 90.0 90.0 60.0				
→	60.0 60.0 60.0 60.0 60.0 90.0 90.0 60.0 60.0				
	FWES~~~~~V	2004-09-22 08:30:00	2004-09-22 08:31:00	0.5	
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	ROHR~~~~~VAV	2004-09-22 08:30:00	2004-09-22 08:31:00	-0.3	
→	0.2 -0.5 0.1 -0.3 0.3 0.1 -0.1 0.3 0.2 0.0 -0.7				
→	-0.6 -0.1 0.0 -0.7 NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	KNOT~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:31:00	0.0	
→	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0				
→	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	ROHR~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:31:00	0.0	
→	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0				
→	0.0 0.0 0.0 0.0 NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	ROHR~~~~~MVEC	2004-09-22 08:30:00	2004-09-22 08:31:00	-2.4	
→	-2.4 2.0 2.0 -4.3 -4.3 0.9 0.9 -6.4 -6.4 2.6 2.6				
→	2.6 2.6 -1.1 -1.1 2.6 2.6 1.6 1.6 0.4 0.4 -6.4				
→	-6.4 -5.3 -5.3 -1.1 -1.1 0.4 0.4 -6.4 -6.4				
	ROHR~~~~~VK	2004-09-22 08:30:00	2004-09-22 08:31:00	-0.3	
→	0.2 -0.5 0.1 -0.3 0.3 0.1 -0.1 0.3 0.2 0.0 -0.7				
→	-0.6 -0.1 0.0 -0.7 NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	PUMP~~~~~QM	2004-09-22 08:30:00	2004-09-22 08:31:00	9.2	
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	KNOT~~~~~PDAMPF	2004-09-22 08:30:00	2004-09-22 08:31:00	0.2	
→	0.7 0.7 0.7 0.2 0.2 0.2 0.2 0.7 0.2 0.2 0.2				
→	0.2 0.7 0.2 0.7 0.2 0.7 0.2 0.2 0.7 0.7 0.7				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	ROHR~~~~~QMI	2004-09-22 08:30:00	2004-09-22 08:31:00	-8.5	
→	7.4 -15.4 3.2 -23.0 9.2 9.2 -3.9 9.2 5.9 1.5 -23.0				
→	-19.1 -3.9 1.5 -23.0 NaN NaN NaN NaN NaN NaN NaN				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	KNOT~~~~~WALTER	2004-09-22 08:30:00	2004-09-22 08:31:00	0.2	
→	0.2 0.1 0.1 0.4 0.2 0.4 0.5 0.0 0.3 0.5 0.5				
→	0.3 0.4 0.0 0.5 0.0 0.8 0.2 0.5 0.3 0.0 1.1				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				
	KNOT~~~~~P	2004-09-22 08:30:00	2004-09-22 08:31:00	3.1	
→	3.2 3.2 3.2 3.0 3.0 3.0 3.2 3.2 3.1				
→	3.0 3.2 3.0 3.2 3.1 3.2 3.1 3.0 3.2 3.2 3.2				
→	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				

(continues on next page)

(continued from previous page)

	FWES~~~~~QM			2004-09-22	08:30:00	2004-09-22	08:31:00	9.2		
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	KNOT~~~~~HMAX_INST			2004-09-22	08:30:00	2004-09-22	08:31:00	2.3		
→	4.0	4.1	4.1	2.0	2.3	2.0	4.3	4.1	2.3	4.3
→	2.1	3.8	2.0	3.8	2.3	3.8	2.3	2.0	3.8	4.1
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	ROHR~~~~~PVECMIN_INST			2004-09-22	08:30:00	2004-09-22	08:31:00	3.1		
→	3.1	3.2	3.2	3.0	3.1	3.2	3.2	3.0	3.0	3.2
→	3.2	3.2	3.1	3.1	3.2	3.2	3.2	3.2	3.2	3.2
→	3.0	3.0	3.0	3.1	3.1	3.2	3.2	3.0	3.0	3.0
	VENT~~~~~QM			2004-09-22	08:30:00	2004-09-22	08:31:00	9.2		
→	9.2	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	KNOT~~~~~PH			2004-09-22	08:30:00	2004-09-22	08:31:00	2.1		
→	2.2	2.2	2.2	2.0	2.0	2.0	2.2	2.2	2.1	2.2
→	2.0	2.2	2.0	2.2	2.1	2.2	2.1	2.0	2.2	2.2
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	FWVB~~~~~W			2004-09-22	08:30:00	2004-09-22	08:31:00	75.8		
→	77.3	61.2	60.7	47.8	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	FWVB~~~~~QM			2004-09-22	08:30:00	2004-09-22	08:31:00	1.9		
→	2.7	1.8	1.5	1.5	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	KNOT~~~~~H			2004-09-22	08:30:00	2004-09-22	08:31:00	2.1		
→	2.2	2.2	2.2	2.0	2.0	2.0	2.2	2.2	2.1	2.2
→	2.0	2.2	2.0	2.2	2.1	2.2	2.1	2.0	2.2	2.2
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	ROHR~~~~~SVEC			2004-09-22	08:30:00	2004-09-22	08:31:00	0.0		
→	88.0	0.0	406.0	0.0	83.6	0.0	88.0	0.0	73.4	0.0
→	0.0	68.6	0.0	109.8	0.0	76.4	0.0	83.6	0.0	164.9
→	195.5	0.0	406.0	0.0	164.9	0.0	109.8	0.0	76.4	0.0
	ROHR~~~~~PVEC			2004-09-22	08:30:00	2004-09-22	08:31:00	3.1		
→	3.1	3.2	3.2	3.0	3.1	3.2	3.2	3.0	3.0	3.2
→	3.2	3.2	3.1	3.1	3.2	3.2	3.2	3.2	3.2	3.2
→	3.0	3.0	3.0	3.1	3.1	3.2	3.2	3.0	3.0	3.0
	ROHR~~~~~QMAV			2004-09-22	08:30:00	2004-09-22	08:31:00	-8.5		
→	7.4	-15.4	3.2	-23.0	9.2	9.2	-3.9	9.2	5.9	1.5
→	-19.1	-3.9	1.5	-23.0	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	ROHR~~~~~VI			2004-09-22	08:30:00	2004-09-22	08:31:00	-0.3		
→	0.2	-0.5	0.1	-0.3	0.3	0.1	-0.1	0.3	0.2	0.0
→	-0.6	-0.1	0.0	-0.7	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	KNOT~~~~~T			2004-09-22	08:30:00	2004-09-22	08:31:00	60.0		
→	90.0	90.0	90.0	60.0	60.0	60.0	60.0	90.0	60.0	60.0
→	60.0	90.0	60.0	90.0	60.0	90.0	60.0	60.0	90.0	90.0
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	VENT~~~~~IAKTIV			2004-09-22	08:30:00	2004-09-22	08:31:00	0.0		
→	0.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	ROHR~~~~~PVECMAX_INST			2004-09-22	08:30:00	2004-09-22	08:31:00	3.3		
→	3.3	5.0	4.8	3.3	3.3	4.8	4.8	3.0	3.0	(continues on next page)
→	5.1	5.1	3.3	3.3	5.1	5.1	4.8	4.8	4.8	4.8
→	3.1	3.1	3.3	3.3	3.3	4.8	4.8	3.0	3.0	

(continued from previous page)

VENT~~~~~V										
→ 0.1	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
KNOT~~~~~HMIN_INST										
→ 2.2	2.2	2.2	2.0	2.0	2.0	2.2	2.2	2.1	2.2	2.0
→ 2.0	2.2	2.0	2.2	2.1	2.2	2.1	2.0	2.2	2.2	2.2
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ROHR~~~~~QMK										
→ 7.4	-15.4	3.2	-23.0	9.2	9.2	-3.9	9.2	5.9	1.5	-23.0
→ -19.1	-3.9	1.5	-23.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
KNOT~~~~~PMIN_INST										
→ 3.2	3.2	3.2	3.0	3.0	3.0	3.2	3.2	3.1	3.2	3.0
→ 3.0	3.2	3.0	3.2	3.1	3.2	3.1	3.0	3.2	3.2	3.2
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
KNOT~~~~~RHO										
→ 965.7	965.7	965.7	983.7	983.7	983.7	983.7	965.7	983.7	983.7	983.7
→ 983.7	965.7	983.7	965.7	983.7	965.7	983.7	983.7	965.7	965.7	965.
→ 7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ROHR~~~~~ZVEC										
→ 20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
→ 20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
→ 20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
KNOT~~~~~PMAX_INST										
→ 5.0	5.1	5.1	3.0	3.3	3.0	5.3	5.1	3.3	5.3	3.0
→ 3.1	4.8	3.0	4.8	3.3	4.8	3.3	3.0	4.8	5.1	4.8
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
KLAP~~~~~V										
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
KLAP~~~~~IAKTIV										
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
PUMP~~~~~IAKTIV										
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
FWVB~~~~~IAKTIV										
→ 0.0	0.0	0.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
FWES~~~~~IAKTIV										
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
TMAX KLAP~~~~~QM										
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ROHR~~~~~RHOVEC										
→ 983.7	965.7	965.7	983.7	983.7	965.7	965.7	983.7	983.7	965.7	965.7
→ 965.7	965.7	983.7	983.7	965.7	965.7	965.7	965.7	965.7	965.7	983.
→ 7	983.7	983.7	983.7	983.7	983.7	965.7	965.7	983.7	983.7	983.7
ROHR~~~~~TVEC										
→ 60.0	90.0	90.0	60.0	60.0	90.0	90.0	60.0	60.0	60.0	60.0
→ 90.0	90.0	60.0	60.0	90.0	90.0	90.0	90.0	90.0	90.0	60.0
→ 60.0	60.0	60.0	60.0	60.0	90.0	90.0	60.0	60.0	60.0	60.0

(continues on next page)

(continued from previous page)

	FWES~~~~~V			2004-09-22	08:30:00	2004-09-22	08:31:00	1.3		
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	ROHR~~~~~VAV			2004-09-22	08:30:00	2004-09-22	08:31:00	-0.1		
→	0.6	-0.2	0.3	-0.1	0.7	0.3	-0.0	0.7	0.5	0.1
→	-0.2	-0.0	0.1	-0.3	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	KNOT~~~~~IAKTIV			2004-09-22	08:30:00	2004-09-22	08:31:00	0.0		
→	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
→	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	ROHR~~~~~IAKTIV			2004-09-22	08:30:00	2004-09-22	08:31:00	0.0		
→	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
→	0.0	0.0	0.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	ROHR~~~~~MVEC			2004-09-22	08:30:00	2004-09-22	08:31:00	-0.9		
→	-0.9	5.3	5.3	-1.6	-1.6	2.4	2.4	-2.6	-2.6	6.4
→	6.4	6.4	-0.4	-0.4	6.4	6.4	4.3	4.3	1.1	1.1
→	-2.6	-2.0	-2.0	-0.4	-0.4	1.1	1.1	-2.6	-2.6	
	ROHR~~~~~VK			2004-09-22	08:30:00	2004-09-22	08:31:00	-0.1		
→	0.6	-0.2	0.3	-0.1	0.7	0.3	-0.0	0.7	0.5	0.1
→	-0.2	-0.0	0.1	-0.3	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	PUMP~~~~~QM			2004-09-22	08:30:00	2004-09-22	08:31:00	23.0		
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	KNOT~~~~~PDAMPF			2004-09-22	08:30:00	2004-09-22	08:31:00	0.2		
→	0.7	0.7	0.7	0.2	0.2	0.2	0.2	0.7	0.2	0.2
→	0.2	0.7	0.2	0.7	0.2	0.7	0.2	0.2	0.7	0.7
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	ROHR~~~~~QMI			2004-09-22	08:30:00	2004-09-22	08:31:00	-3.2		
→	19.1	-5.9	8.5	-9.2	23.0	23.0	-1.5	23.0	15.4	3.9
→	-7.4	-1.5	3.9	-9.2	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	KNOT~~~~~WALTER			2004-09-22	08:30:00	2004-09-22	08:31:00	0.5		
→	0.4	0.2	0.1	1.0	0.5	1.1	1.2	0.0	0.8	1.2
→	0.8	1.0	0.0	1.2	0.0	2.2	0.7	1.2	0.9	0.0
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	KNOT~~~~~P			2004-09-22	08:30:00	2004-09-22	08:31:00	3.3		
→	5.0	5.1	5.1	3.0	3.3	3.0	5.3	5.1	3.3	5.3
→	3.1	4.8	3.0	4.8	3.3	4.8	3.3	3.0	4.8	5.1
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	FWES~~~~~QM			2004-09-22	08:30:00	2004-09-22	08:31:00	23.0		
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	KNOT~~~~~HMAX_INST			2004-09-22	08:30:00	2004-09-22	08:31:00	2.3		
→	4.0	4.1	4.1	2.0	2.3	2.0	4.3	4.1	2.3	4.3
→	2.1	3.8	2.0	3.8	2.3	3.8	2.3	2.0	3.8	4.1
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		
	ROHR~~~~~PVECMIN_INST			2004-09-22	08:30:00	2004-09-22	08:31:00	3.3		
→	3.3	5.0	4.8	3.3	3.3	4.8	4.8	3.0	3.0	5.1
→	5.1	5.1	3.3	3.3	5.1	5.1	4.8	4.8	4.8	3.0
→	3.1	3.1	3.3	3.3	3.3	4.8	4.8	3.0	3.0	
	VENT~~~~~QM			2004-09-22	08:30:00	2004-09-22	08:31:00	23.0		
→	23.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
→	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		

(continues on next page)

(continued from previous page)

KNOT~~~~~PH											
→ 4.0	4.1	4.1	2.0	2.3	2.0	4.3	4.1	2.3	4.3	2.0	↵
→ 2.1	3.8	2.0	3.8	2.3	3.8	2.3	2.0	3.8	4.1	3.8	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		↵
FWVB~~~~~W											
→ 200.0	160.0	160.0	120.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
FWVB~~~~~QM											
→ 6.9	4.6	3.9	3.7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
KNOT~~~~~H											
→ 4.0	4.1	4.1	2.0	2.3	2.0	4.3	4.1	2.3	4.3	2.0	↵
→ 2.1	3.8	2.0	3.8	2.3	3.8	2.3	2.0	3.8	4.1	3.8	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		↵
ROHR~~~~~SVEC											
→ 88.0	0.0	406.0	0.0	83.6	0.0	88.0	0.0	73.4	0.0	195.5	↵
→ 0.0	68.6	0.0	109.8	0.0	76.4	0.0	83.6	0.0	164.9	0.0	↵
→ 195.5	0.0	406.0	0.0	164.9	0.0	109.8	0.0	76.4			↵
ROHR~~~~~PVEC											
→ 3.3	5.0	4.8	3.3	3.3	4.8	4.8	3.0	3.0	5.1	5.0	↵
→ 5.1	5.1	3.3	3.3	5.1	5.1	4.8	4.8	4.8	4.8	3.0	↵
→ 3.1	3.1	3.3	3.3	3.3	4.8	4.8	3.0	3.0			↵
ROHR~~~~~QMAV											
→ 19.1	-5.9	8.5	-9.2	23.0	23.0	-1.5	23.0	15.4	3.9	-9.2	↵
→ -7.4	-1.5	3.9	-9.2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		↵
ROHR~~~~~VI											
→ 0.6	-0.2	0.3	-0.1	0.7	0.3	-0.0	0.7	0.5	0.1	-0.3	↵
→ -0.2	-0.0	0.1	-0.3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		↵
KNOT~~~~~T											
→ 90.0	90.0	90.0	60.0	60.0	60.0	60.0	90.0	60.0	60.0	60.0	↵
→ 60.0	90.0	60.0	90.0	60.0	90.0	60.0	60.0	90.0	90.0	90.0	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		↵
VENT~~~~~IAKTIV											
→ 0.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		↵
ROHR~~~~~PVECMAX_INST											
→ 3.3	5.0	4.8	3.3	3.3	4.8	4.8	3.0	3.0	5.1	5.0	↵
→ 5.1	5.1	3.3	3.3	5.1	5.1	4.8	4.8	4.8	4.8	3.0	↵
→ 3.1	3.1	3.3	3.3	3.3	4.8	4.8	3.0	3.0			↵
VENT~~~~~V											
→ 0.4	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		↵
KNOT~~~~~HMIN_INST											
→ 4.0	4.1	4.1	2.0	2.3	2.0	4.3	4.1	2.3	4.3	2.0	↵
→ 2.1	3.8	2.0	3.8	2.3	3.8	2.3	2.0	3.8	4.1	3.8	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		↵
ROHR~~~~~QMK											
→ 19.1	-5.9	8.5	-9.2	23.0	23.0	-1.5	23.0	15.4	3.9	-9.2	↵
→ -7.4	-1.5	3.9	-9.2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN		↵
KNOT~~~~~PMIN_INST											
→ 5.0	5.1	5.1	3.0	3.3	3.0	5.3	5.1	3.3			↵
→ 3.1	4.8	3.0	4.8	3.3	4.8	3.3	3.0	4.8	5.1	4.8	↵
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN			↵

(continues on next page)

(continued from previous page)

	KNOT~~~~~RHO	2004-09-22 08:30:00	2004-09-22 08:31:00	983.7	↵
↵	965.7 965.7 965.7 983.7 983.7 983.7 983.7 965.7 983.7 983.7 983.7				↵
↵	983.7 965.7 983.7 965.7 983.7 965.7 983.7 983.7 965.7 965.7 965.				↵
↵	7 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
	ROHR~~~~~ZVEC	2004-09-22 08:30:00	2004-09-22 08:31:00	20.0	↵
↵	20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0				↵
↵	20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0				↵
↵	20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0				↵
	KNOT~~~~~PMAX_INST	2004-09-22 08:30:00	2004-09-22 08:31:00	3.3	↵
↵	5.0 5.1 5.1 3.0 3.3 3.0 5.3 5.1 3.3 5.3 3.0				↵
↵	3.1 4.8 3.0 4.8 3.3 4.8 3.3 3.0 4.8 5.1 4.8				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
	KLAP~~~~~V	2004-09-22 08:30:00	2004-09-22 08:31:00	1.3	↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
	KLAP~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:31:00	0.0	↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
	PUMP~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:31:00	0.0	↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
	FWVB~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:31:00	0.0	↵
↵	0.0 0.0 0.0 0.0 NaN NaN NaN NaN NaN NaN NaN				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
	FWES~~~~~IAKTIV	2004-09-22 08:30:00	2004-09-22 08:31:00	0.0	↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵
↵	NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN				↵

_checkMxsVecsFile (*fullCheck=False*)

Returns (firstTime,lastTime,NOfTimes).

Args:

- **fullCheck (bool)**
 - False (default): only 1st and last h5Keys are read
 - True: all h5Keys are read

Returns:

- (firstTime,lastTime,NOfTimes)
- if self.h5FileVecs does not exist None,None,[] is returned

Raises: MxError**_handleMxsVecsFileDeletion** (*mxsFile, newMxsVecsFile=False*)

Handles the deletion of mxsVecsFile.

Args:

- mxsFile
- newMxsVecsFile (default: False)

mxsVecsFile is DELETED! if:

- existing and older than mxsFile
- or newMxsVecsFile is True

Raises: MxError

setResultsToMxsFile (*mxsFile=None, add=False, NewH5Vec=False, maxRecords=None*)

Sets (default) or adds mxsFile-Content to .df.

Args:

- **mxsFile (str)**
 - None (default): .mxsFile is used
- add (bool): default: False: sets df to mxsFile-Content
- NewH5Vec: False (default); if True, an existing mxsVecsFile will be deleted even if it is newer than mxsFile
- maxRecords: Anzahl der max. zu lesenden Records

Returns:

- timesWrittenToMxsVecs

.df

- index: TIMESTAMP
- self.df.index.is_unique will be True
- because in SIR 3S'
 - **+TIME** is dropped (STAT is used in df)
 - +TMIN/TMAX are dropped (not used in df)
- and because resulting overlapping TIMESTAMPS due to intersections (add=True) are also dropped

.h5FileVecs

- is updated with mxsFile-Content
- **is DELETED! before if existing and**
 - older than mxsFile
 - or newMxsVecsFile

Raises: MxError

```
>>> mxDummy=mxs['GPipes']
>>> try:
...     mx=Mx(mx1File=mxDummy.mx1File,NoH5Read=True,NoMxsRead=True) # avoid
↳doing anything than just plain Init
... except:
...     from Mx import Mx
...     mx=Mx(mx1File=mxDummy.mx1File,NoH5Read=True,NoMxsRead=True) # avoid
↳doing anything than just plain Init
>>> mx.df
```

(continues on next page)

(continued from previous page)

```
>>> mx.setResultsToMxsFile(NewH5Vec=True)
1
```

setResultsToMxsZipFile (*mxsZipFile=None, add=False, NewH5Vec=False, maxRecords=None*)
Sets (default) or adds mxsZipFile-Content to .df.

Args:

- **mxsZipFile (str)**
 - None (default): .mxsFile is used
- add (bool): default: False: sets df to mxsZipFile-Content
- NewH5Vec
- maxRecords: Anzahl der max. zu lesenden Records

Returns:

- timesWrittenToMxsVecsFromZip

Raises: MxError

ToH5 (*h5File=None*)

Stores .mx1Df, .mx2Df, .df, .dfVecAggs to h5File.

Args: h5File(str): default: None: self.h5File is used

.h5File

- is !DELETED! before if existing

Keys used

- /MX1
- /MX2
- /MXS

Raises: MxError

```
# -q -m 0 -s ToH5 -t both -y yes -z no -p yes -w LocalHeatingNetwork >>>
mx=mxs['LocalHeatingNetwork'] >>> mx.ToH5() >>> nOfRecs=mx.setResultsToMxsFile() >>>
nOfRecs 5 >>> mx.ToH5()
```

FromH5 (*h5File=None*)

Sets .mx1Df, .mx2Df, .df, .dfVecAggs to h5File-Content.

Args:

- **h5File(str)**
 - None (default): .h5File is used

Keys processed:

- /MX1
- /MX2
- /MXS
- /VecAggs

Raises: MxError

```
# -q -m 0 -s FromH5 -t both -y no -z yes -w LocalHeatingNetwork >>> mx=mxs['LocalHeatingNetwork']
>>> mx.FromH5()
```

getMxsVecsFileData (*timesReq=None, fastMode=False*)

Returns List of dfs with mxsVecsFileData. One TIMESTAMP (index) per df.

Args:

- **timesReq: List of TIMESTAMPS**
 - if None: a List with a single time only, the 1st Time, is constructed as timesReq
- fastMode (default: False): H5-Access with no Checks

Returns:

- List of dfs with mxsVecsFileData
- empty List if no TIMESTAMP could be found
- one df per TIMESTAMP
- index: TIMESTAMP

Raises: MxError

```
>>> mx=mxs['LocalHeatingNetwork']
>>> mx.delFiles()
>>> mx.setResultsToMxsFile() # reads TIMESTAMPS and constructs .vec.h5 while_
↳reading; returns TIMESTAMPS read
5
>>> mxVecsFileDataLst=mx.getMxsVecsFileData()
>>> len(mxVecsFileDataLst)
1
>>> mxVecsFileData=mxVecsFileDataLst[0]
>>> type(mxVecsFileData)
<class 'pandas.core.frame.DataFrame'>
>>> mxVecsFileData.index[0]
Timestamp('2004-09-22 08:30:00+0000', tz='UTC')
>>> vecsFileDataOneCol=mxVecsFileData['ROHR~*~*~*~SVEC']
>>> vecsFileDataOneColResult=vecsFileDataOneCol[0]
>>> vecsFileDataOneColResult[-1]
76.4000015258789
>>> mxVecsFileDataLst=mx.getMxsVecsFileData(fastMode=True)
>>> mxVecsFileData=mxVecsFileDataLst[0]
>>> vecsFileDataOneCol=mxVecsFileData['ROHR~*~*~*~SVEC']
>>> vecsFileDataOneColResult=vecsFileDataOneCol[0]
>>> vecsFileDataOneColResult[-1]
76.4000015258789
>>> import pandas as pd
>>> timeNotAva=mx.df.index[-1]+pd.Timedelta('666 milliseconds')
>>> timeNotAva
Timestamp('2004-09-22 08:31:00.666000+0000', tz='UTC')
>>> mxVecsFileDataLst=mx.getMxsVecsFileData(timesReq=[timeNotAva])
>>> len(mxVecsFileDataLst)
0
```

unPackMxsVecsFileDataDf (*mxVecsFileData, mIndex, returnMultiIndex=True*)

Unpacks mxVecsFileData-Content into a returned df.

Args:

- **mxVecsFileData:** the stuff returned by `getMxsVecsFileData` is: List of dfs with `mxsVecsFileData`. One **TIMESTAMP** (index) per df. One of these dfs.
- **mIndex: MultiIndex to be used**
 - 1st Level: Timestamps
 - 2nd Level: cols (Sir3sIds to be unpacked)
- **returnMultiIndex (default: True):**
 - if True, the data is unpacked row-wise (stored in cols) and the index is mIndex and the col-Labels are mx2Idx
 - if False, the data is unpacked col-wise (stored in rows) and the index is mx2Idx and th col-Labels are the cols (the Sir3sIds)

Returns:

- df

Raises: MxError

```

>>> mx=mxs['LocalHeatingNetwork']
>>> mx.delFiles()
>>> mx.setResultsToMxsFile() # reads 5 TIMESTAMPS and constructs .vec.h5
↳while reading
5
>>> timesReq=list(mx.df.index[:]) # all times
>>> mxVecsFileDataLst=mx.getMxsVecsFileData(timesReq=timesReq)
>>> len(mxVecsFileDataLst)
5
>>> mxVecsFileData=mxVecsFileDataLst[0]
>>> # construct MultiIndex Start ... ---
>>> colsToBeUnpacked=['ROHR~*~*~*~SVEC','ROHR~*~*~*~QMAV','KNOT~*~*~*~PH'] #
↳mxVecsFileData.columns.tolist() for all columns
>>> arrays=[mxVecsFileData.index[0]]*len(colsToBeUnpacked),colsToBeUnpacked]
>>> tuples = list(zip(*arrays))
>>> import pandas as pd
>>> mIndex = pd.MultiIndex.from_tuples(tuples, names=['Timestamp', 'Sir3sID'])
>>> mIndex
MultiIndex([(2004-09-22 08:30:00+00:00', 'ROHR~*~*~*~SVEC'),
            (2004-09-22 08:30:00+00:00', 'ROHR~*~*~*~QMAV'),
            (2004-09-22 08:30:00+00:00', 'KNOT~*~*~*~PH')],
            names=['Timestamp', 'Sir3sID'])
>>> # construct MultiIndex End ... ---
>>> df=mx.unPackMxsVecsFileDataDf(mxVecsFileData,mIndex,
↳returnMultiIndex=False)
>>> df.round(1)

```

	ROHR~*~*~*~SVEC	ROHR~*~*~*~QMAV	KNOT~*~*~*~PH
0	0.0	-8.5	2.3
1	88.0	19.1	4.0
2	0.0	-15.4	4.1
3	406.0	8.5	4.1
4	0.0	-23.0	2.0
5	83.6	23.0	2.3
6	0.0	23.0	2.0
7	88.0	-3.9	4.3
8	0.0	23.0	4.1
9	73.4	15.4	2.3
10	0.0	3.9	4.3

(continues on next page)

```

11      195.5      -23.0      2.0
12      0.0      -19.1      2.1
13      68.6      -3.9      3.8
14      0.0      3.9      2.0
15      109.8      -23.0      3.8
16      0.0      NaN      2.3
17      76.4      NaN      3.8
18      0.0      NaN      2.3
19      83.6      NaN      2.0
20      0.0      NaN      3.8
21      164.9      NaN      4.1
22      0.0      NaN      3.8
23      195.5      NaN      NaN
24      0.0      NaN      NaN
25      406.0      NaN      NaN
26      0.0      NaN      NaN
27      164.9      NaN      NaN
28      0.0      NaN      NaN
29      109.8      NaN      NaN
30      0.0      NaN      NaN
31      76.4      NaN      NaN
>>> df=mx.unpackMxsVecsFileDataDf(mxVecsFileData,mIndex)
>>> df.round(1)

```

	0	1	2	3	4	5
→ 6	7	8	9	10	11	12
→ 19	20	21	22	23	24	25
Timestamp	Sir3sID					
→						
→						
2004-09-22 08:30:00+00:00	ROHR~***~SVEC	0.0	88.0	0.0	406.0	0.0
→ 0.0	88.0	0.0	73.4	0.0	195.5	0.0
→ 83.6	0.0	164.9	0.0	195.5	0.0	406.0
	ROHR~***~QMAV	-8.5	19.1	-15.4	8.5	-23.0
→ 23.0	-3.9	23.0	15.4	3.9	-23.0	-19.1
→ NaN	NaN	NaN	NaN	NaN	NaN	NaN
	KNOT~***~PH	2.3	4.0	4.1	4.1	2.0
→ 2.0	4.3	4.1	2.3	4.3	2.0	2.1
→ 2.0	3.8	4.1	3.8	NaN	NaN	NaN

```

>>> dfs=[]
>>> for idx,mxVecsFileData in enumerate(mxVecsFileDataLst):
...     arrays=[[mxVecsFileData.index[0]]*len(colsToBeUnpacked),
→colsToBeUnpacked]
...     tuples = list(zip(*arrays))
...     mIndex = pd.MultiIndex.from_tuples(tuples, names=['Timestamp',
→'Sir3sID'])
...     dfs.append(mx.unpackMxsVecsFileDataDf(mxVecsFileData,mIndex))
>>> df=pd.concat(dfs)
>>> idx=pd.IndexSlice
>>> dfOneVecChannel=df.loc[(idx[:],'KNOT~***~PH'),0:22] # df.loc[(idx[:],
→idx[:]),idx[:]: everything
>>> dfOneVecChannel.round(1)

```

	0	1	2	3	4	5	6
→ 7	8	9	10	11	12	13	14
→ 22	15	16	17	18	19	20	21
Timestamp	Sir3sID						
→							
→							
2004-09-22 08:30:00+00:00	KNOT~***~PH	2.3	4.0	4.1	4.1	2.0	2.3
→ 3	4.1	2.3	4.3	2.0	2.1	3.8	2.0

(continues on next page)

(continued from previous page)

```

2004-09-22 08:30:15+00:00 KNOT~*~*~*~*~PH 2.3 3.0 3.1 3.2 2.0 2.3 2.0 3.
↪3 3.2 2.3 3.3 2.0 2.1 2.9 2.0 2.9 2.3 2.9 2.3 2.0 2.9 3.2 2.9
2004-09-22 08:30:30+00:00 KNOT~*~*~*~*~PH 2.1 2.5 2.6 2.6 2.0 2.1 2.0 2.
↪7 2.6 2.1 2.7 2.0 2.1 2.5 2.0 2.5 2.1 2.5 2.1 2.0 2.5 2.6 2.5
2004-09-22 08:30:45+00:00 KNOT~*~*~*~*~PH 2.1 2.2 2.2 2.2 2.0 2.0 2.0 2.
↪2 2.2 2.1 2.2 2.0 2.0 2.2 2.0 2.2 2.1 2.2 2.1 2.0 2.2 2.2 2.2
2004-09-22 08:31:00+00:00 KNOT~*~*~*~*~PH 2.3 4.0 4.1 4.1 2.0 2.3 2.0 4.
↪3 4.1 2.3 4.3 2.0 2.1 3.8 2.0 3.8 2.3 3.8 2.3 2.0 3.8 4.1 3.8
>>> dfOneVecChannel.min().round(1)
0      2.1
1      2.2
2      2.2
3      2.2
4      2.0
5      2.0
6      2.0
7      2.2
8      2.2
9      2.1
10     2.2
11     2.0
12     2.0
13     2.2
14     2.0
15     2.2
16     2.1
17     2.2
18     2.1
19     2.0
20     2.2
21     2.2
22     2.2
dtype: float64
>>> df.min(level=1).round(1)
      0      1      2      3      4      5      6      7      8      9
↪10    11    12    13    14    15    16    17    18    19    20    21    22
↪23    24    25    26    27    28    29    30    31
Sir3sID
↪
↪
ROHR~*~*~*~*~SVEC 0.0 88.0 0.0 406.0 0.0 83.6 0.0 88.0 0.0 73.4 0.
↪0 195.5 0.0 68.6 0.0 109.8 0.0 76.4 0.0 83.6 0.0 164.9 0.0
↪195.5 0.0 406.0 0.0 164.9 0.0 109.8 0.0 76.4
ROHR~*~*~*~*~QMAV -8.5 7.4 -15.4 3.2 -23.0 9.2 9.2 -3.9 9.2 5.9 1.
↪5 -23.0 -19.1 -3.9 1.5 -23.0 NaN NaN NaN NaN NaN NaN NaN
↪NaN NaN NaN NaN NaN NaN NaN NaN NaN
KNOT~*~*~*~*~PH 2.1 2.2 2.2 2.2 2.0 2.0 2.0 2.2 2.2 2.1 2.
↪2 2.0 2.0 2.2 2.0 2.2 2.1 2.2 2.1 2.0 2.2 2.2 2.2
↪NaN NaN NaN NaN NaN NaN NaN NaN NaN

```

getVecAggs (*time1st=None*, *time1stIncluded=True*, *time2nd=None*, *time2ndIncluded=True*, *aTIME=False*)

Gets (or calcs) Aggregates (MIN, MAX, ...) of mxsVecsFileData between the 2 Times.

- New calced Aggregates are stored in dfVecAggs.

Args:

- time1st: TIMESTAMP (first if None)
- time2nd: TIMESTAMP (last if None)
- time1stIncluded
- time2ndIncluded
- aTIME: if true, time1st is considered to be a TIME to be fetched; time2nd is ignored

Returns:

- **dfs with MultiIndex:**
 - Level 0: 'MIN', 'MAX', ...
 - Level 1: col (Sir3sID)
- [* the 2 Time Idx (Level 2 and 3) are dropped] * cols: mx2Idx
- timeL: left ScenTimeStamp included in calculating the Aggregate
- timeR: right ScenTimeStamp included in calculating the Aggregate

Raises: MxError

```
# -q -m 0 -s getVecAggs -t both -y yes -z no -p yes -w LocalHeatingNetwork >>>
mx=mxs['LocalHeatingNetwork'] >>> mx.delFiles() >>> mx.setResultsToMxsFile() # reads 5
TIMESTAMPS and constructs .vec.h5 while reading 5 >>> # check dfVecAggs to demonstrate how
getVecAggs stores to / reads from getVecAggs >>> Sir3sIDs=mx.dfVecAggs.index.unique(level=1).values
>>> len(Sir3sIDs) 41 >>> mx.dfVecAggs.index.unique(level=0).values array(['TIME',
'TMIN', 'TMAX'], dtype=object) >>> len(mx.dfVecAggs.columns.tolist()) 32 >>>
mx.dfVecAggs.shape # (3*41,32) (123, 32) >>> df,tL,tR=mx.getVecAggs() >>>
mx.dfVecAggs.index.unique(level=0).values array(['TIME', 'TMIN', 'TMAX', 'MIN', 'MAX',
'DIF'], dtype=object) >>> mx.dfVecAggs.shape (246, 32) >>> import pandas as pd >>>
#idx=pd.IndexSlice >>> df.loc[(['MIN','MAX','DIF'],'KNOT~*~*~PH'),0:22].round(1) ##
df.loc[(slice(None),'KNOT~*~*~PH'),slice(None)] # df.loc[(idx[:],'KNOT~*~*~PH'),idx[:]]

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

TYPE Sir3sID MIN KNOT~*~*~PH 2.1 2.2 2.2 2.2 2.0 2.0 2.0 2.2 2.2 2.1 2.2 2.0 2.0 2.2 2.0 2.2 2.1 2.2
2.1 2.0 2.2 2.2 2.2 MAX KNOT~*~*~PH 2.3 4.0 4.1 4.1 2.0 2.3 2.0 4.3 4.1 2.3 4.3 2.0 2.1 3.8 2.0 3.8
2.3 3.8 2.3 2.0 3.8 4.1 3.8 DIF KNOT~*~*~PH 0.0 -0.0 -0.0 -0.0 0.0 0.0 0.0 -0.0 -0.0 0.0 -0.0 0.0 0.0
-0.0 0.0 -0.0 0.0 -0.0 0.0 0.0 -0.0 -0.0 -0.0 >>> # demonstrate how to transform an getVecAggs()-df-Result
for Xm... >>> dfT=df.loc[(['MIN',df.index.get_level_values(1).tolist()),:].transpose(copy=True) ##
dfT=df.loc[(['MIN',slice(None)),:].transpose(copy=True) # dfT=df.loc[(['MIN',idx[:]),idx[:]].transpose(copy=True)
>>> colIndex=dfT.columns.droplevel(level=0) >>> colIndex.name=None >>>
pd.DataFrame(dfT.values,columns=colIndex)[['ROHR~*~*~SVEC', 'ROHR~*~*~QMAV',
'KNOT~*~*~PH']].round(1)

ROHR~*~*~SVEC ROHR~*~*~QMAV KNOT~*~*~PH

0 0.0 -8.5 2.1 1 88.0 7.4 2.2 2 0.0 -15.4 2.2 3 406.0 3.2 2.2 4 0.0 -23.0 2.0 5 83.6 9.2 2.0 6 0.0 9.2 2.0 7
88.0 -3.9 2.2 8 0.0 9.2 2.2 9 73.4 5.9 2.1 10 0.0 1.5 2.2 11 195.5 -23.0 2.0 12 0.0 -19.1 2.0 13 68.6 -3.9 2.2
14 0.0 1.5 2.0 15 109.8 -23.0 2.2 16 0.0 NaN 2.1 17 76.4 NaN 2.2 18 0.0 NaN 2.1 19 83.6 NaN 2.0 20 0.0
NaN 2.2 21 164.9 NaN 2.2 22 0.0 NaN 2.2 23 195.5 NaN NaN 24 0.0 NaN NaN 25 406.0 NaN NaN 26 0.0
NaN NaN 27 164.9 NaN NaN 28 0.0 NaN NaN 29 109.8 NaN NaN 30 0.0 NaN NaN 31 76.4 NaN NaN
>>> df,tL,tR=mx.getVecAggs() >>> mx.dfVecAggs.shape (246, 32) >>> # demonstrate how to transform
an AggEntry for Xm... >>> df=mx.dfVecAggs.loc[(['TMIN',slice(None),mx.df.index[0],mx.df.index[-
1]),:] >>> dfT=df.transpose(copy=True) >>> colIndex=dfT.columns.droplevel(level=0) >>> colIn-
dex=colIndex.droplevel(level=1) >>> colIndex=colIndex.droplevel(level=1) >>> colIndex.name=None
```

```
>>> pd.DataFrame(dfT.values,columns=colIndex)[['ROHR~*~*~*~SVEC', 'ROHR~*~*~*~QMAV',
'KNOT~*~*~*~PH']].round(1)
```

```
ROHR~*~*~*~SVEC ROHR~*~*~*~QMAV KNOT~*~*~*~PH
```

```
0 0.0 -8.5 2.1 1 88.0 7.4 2.2 2 0.0 -15.4 2.2 3 406.0 3.2 2.2 4 0.0 -23.0 2.0 5 83.6 9.2 2.0 6 0.0 9.2 2.0 7
88.0 -3.9 2.2 8 0.0 9.2 2.2 9 73.4 5.9 2.1 10 0.0 1.5 2.2 11 195.5 -23.0 2.0 12 0.0 -19.1 2.0 13 68.6 -3.9
2.2 14 0.0 1.5 2.0 15 109.8 -23.0 2.2 16 0.0 NaN 2.1 17 76.4 NaN 2.2 18 0.0 NaN 2.1 19 83.6 NaN 2.0 20
0.0 NaN 2.2 21 164.9 NaN 2.2 22 0.0 NaN 2.2 23 195.5 NaN NaN 24 0.0 NaN NaN 25 406.0 NaN NaN
26 0.0 NaN NaN 27 164.9 NaN NaN 28 0.0 NaN NaN 29 109.8 NaN NaN 30 0.0 NaN NaN 31 76.4 NaN
NaN
```

dumpInMxsFormat (*mxsDumpFile=None*)

Dumps in MXS-Format to mxsDumpFile (for testing purposes).

Returns:

- **(TimeStampsDumped, TimeStampsFoundInH5)**
 - normally: TimeStampsDumped=TimeStampsFoundInH5
 - if TimesStamps in self.df are manipulated ...
 - ... the H5-Content remains unchanged
 - in effect the H5-Content can be different from self.df-Content ...
 - während in self.df die Zeiten (Index) immer geordnet und voneinander verschieden sind
 - sind beim H5-Content nur die Keys voneinander verschieden
 - um pruefen zu koennen, ob alle Zeiten in self.df im H5-Content auch gefunden wurden, wird TimeStampsFoundInH5 mit ausgegeben
 - gedumped werden immer alle Zeiten aus self.df
 - * fuer jede im H5-Content nicht gefundene Zeit wird das Ergebnis der zuletzt zuvor gefundenen Zeit ausgegeben

Raises: MxError

__getMx1DfAsOneString (*regex='KNOT~\S*~\S*~\d+~[P|Q]{1}[H|M]{1}\$'*)

Returns .mx1Df-Content as one String (for Doctest-Purposes).

Raises: MxError

__getDfAsOneString (*regex='KNOT~\S*~\S*~\d+~[P|Q]{1}[H|M]{1}\$'*)

Returns .df-Content as one String (for Doctest-Purposes).

Raises: MxError

__getDfVecAsOneString (*df=None, regex='KNOT~\S*~\S*~\S*~[P|Q]{1}[H|M]{1}\$'*)

Returns dfVec-Content as one String (for Doctest-Purposes).

Raises: MxError

getSir3sIDFromSir3sIDoPK (*sir3sIDoPK=""*)

Returns Sir3sID from Sir3sIDoPK.

Raises: MxError

Returns: Sir3sID in mx.df (a colname from mx.df) the 1st match is returned None, if no corresponding Sir3sID found

```
# -q -m 0 -s getSir3sIDFromSir3sIDoPK -t both -y yes -z no -w LocalHeatingNetwork
>>> mx=mxs['LocalHeatingNetwork'] >>> sir3sIDoPK='ALLG~~~LINEPACKGEOM' >>>
```

```
mx.getSir3sIDFromSir3sIDoPK(sir3sIDoPK) 'ALLG~~~4639827058859487185~LINEPACKGEOM'  
>>> mx.getSir3sIDFromSir3sIDoPK('666')
```

Mx.getMicrosecondsFromRefTime (*refTime*, *time*)

Returns time in microseconds since refTime.

Args:

- refTime
- time

Raises: MxError

```
>>> import pandas as pd  
>>> timeReadFromMx=b'2019-01-01 00:00:12.500000 '  
>>> timeRefMx=b'2019-01-01 00:00:00.000000 '  
>>> timeStampTimeReadFromMx=pd.to_datetime(timeReadFromMx.decode(),utc=True)  
>>> timeStampTimeRefFromMx=pd.to_datetime(timeRefMx.decode(),utc=True)  
>>> timeDelta=timeStampTimeReadFromMx-timeStampTimeRefFromMx  
>>> timeDelta.total_seconds()  
12.5  
>>> import Mx  
>>> Mx.getMicrosecondsFromRefTime(timeStampTimeRefFromMx,timeStampTimeReadFromMx)  
12500
```

exception Mx.**MxError** (*value*)

MxError.


```
>>> # ---
>>> # Imports
>>> # ---
>>> import os
>>> import logging
>>> logger = logging.getLogger('PT3S.Xm')
>>> # ---
>>> # path
>>> # ---
>>> if __name__ == "__main__":
...     try:
...         dummy=__file__
...         logger.debug("{0:s}{1:s}{2:s}".format('DOCTEST: __main__ Context: ', 'path = ',
↳os.path.dirname(__file__)', " ."))
...         path = os.path.dirname(__file__)
...     except NameError:
...         logger.debug("{0:s}{1:s}{2:s}".format('DOCTEST: __main__ Context: ', "path =
↳'.' because __file__ not defined: ", "from Xm import Xm follows ..."))
...         path = '.'
...         from Xm import Xm
...     else:
...         logger.debug("{0:s}{1:s}{2:s}{3:s}".format('DOCTEST: Not __main__ Context: ', '_
↳__name__: ', __name__, "path = '.'"))
...         path = '.'
>>> try:
...     from PT3S import Mx
... except ImportError:
...     logger.debug("{0:s}{1:s}".format("DOCTEST: ImportError: from PT3S import Mx: ",
↳"- trying import Mx instead ... maybe pip install -e . is active ..."))
...     import Mx
>>> # ---
>>> # testDir
>>> # ---
>>> # globs={'testDir':'testdata'}
>>> try:
...     dummy= testDir
... except NameError:
...     testDir='testdata'
>>> import pandas as pd
>>> # ---
>>> # Clean Up
>>> # ---
>>> h5File=os.path.join(os.path.join(path, testDir), 'OneLPipe.h5')
>>> if os.path.exists(h5File):
```

(continues on next page)

(continued from previous page)

```

...     os.remove(h5File)
>>> # ---
>>> # Init
>>> # ---
>>> xmlFile=os.path.join(os.path.join(path,testDir),'OneLPipe.XML')
>>> xm=Xm(xmlFile=xmlFile)
>>> # ---
>>> # a View
>>> # ---
>>> v='vKNOT'
>>> v in xm.dataFrames
True
>>> isinstance(xm.dataFrames[v],pd.core.frame.DataFrame)
True
>>> # ---
>>> # ToH5
>>> # ---
>>> xm.ToH5()
>>> os.path.exists(xm.h5File)
True
>>> # ---
>>> # force Read H5 instead of Xml
>>> # ---
>>> os.rename(xm.xmlFile,xm.xmlFile+'.blind')
>>> xm=Xm(xmlFile=xmlFile)
>>> os.rename(xm.xmlFile+'.blind',xm.xmlFile)
>>> # ---
>>> vKNOT=xm.dataFrames['vKNOT']
>>> vStr=xm.getVersion(type='BZ')
>>> import re
>>> m=re.search('Sir(?:P<Db3s>[DBdb3Ss]{2})-(?:P<Major>\d+)-(?:P<Minor>\d+)$',vStr) # i.
↳ e. Sir3S-90-10
>>> minorVer=int(m.group('Minor'))
>>> # minorVer
>>> if minorVer>=12:
...     shapeSet=(2,40)
... else:
...     shapeSet=(2,40)
>>> shapeSet == vKNOT[(vKNOT.KTYP.isin(['QKON','PKON'])) & (vKNOT.BESCHREIBUNG.fillna(
↳ '').str.startswith('Template Element')==False)].shape
True
>>> vROHR=xm.dataFrames['vROHR']
>>> vROHR.shape
(1, 74)
>>> isinstance(vROHR['pXCors'],pd.core.series.Series)
True
>>> vROHR['pXCors'][0]
[0.0, 500.0]
>>> vROHR.pYCors[0]
[0.0, 0.0]
>>> # ---
>>> # getWDirModelDirModelName()
>>> # ---
>>> (wDir,modelDir,modelName,mx1File)=xm.getWDirModelDirModelName()
>>> modelName
'M-1-0-1'
>>> # ---

```

(continues on next page)

(continued from previous page)

```

>>> # H5-Deletion if NoH5Read=True
>>> # ---
>>> if os.path.exists(xm.h5File):
...     os.remove(xm.h5File)
>>> xm=Xm(xmlFile=xmlFile)
>>> xm.ToH5()
>>> os.path.exists(xm.h5File)
True
>>> xm=Xm(xmlFile=xmlFile,NoH5Read=True)
>>> os.path.exists(xm.h5File)
False
>>> # ---
>>> # print-Options
>>> # ---
>>> pd.set_option('display.max_columns',None)
>>> pd.set_option('display.max_rows',None)
>>> pd.set_option('display.max_colwidth',666666)
>>> pd.set_option('display.width',666666666)
>>> # ---
>>> # vKNOT
>>> # ---
>>> print(xm._getvXXXXAsOneString(vXXXX='vKNOT',dropColList=['LFKT_NAME','LF','LF_min',
↳ 'LF_max','PVAR_NAME','PH','PH_min','PH_max','PZON_NAME','FSTF_NAME','STOF_NAME',
↳ 'GMIX_NAME','UTMP_NAME','2L_NAME','2L_KVR','fkHYDR','fkFQPS']))
    NAME BESCHREIBUNG          IDREFERENZ      CONT CONT_ID CONT_LFDNR CONT_VKNO
↳ KTYP LFAKT      QM_EIN QVAR_NAME  QM  QM_min  QM_max KVR   TE   TM XKOR YKOR ZKOR
↳
↳          pk                      tk pXCor  pYCor
0      I          NaN  3S5642914844465475844  OneLPipe    1001          NaN          NaN
↳ QKON      1  176.7146          NaN NaN          NaN          0  NaN  10  300  600  10
↳ 5642914844465475844  5642914844465475844    0.0    0.0
1      K          NaN  3S5289899964753656852  OneLPipe    1001          NaN          NaN
↳ PKON      1          0          NaN NaN          NaN          0  NaN  10  800  600  10
↳ 5289899964753656852  5289899964753656852  500.0    0.0
>>> # ---
>>> # vROHR
>>> # ---
>>> print(xm._getvXXXXAsOneString(vXXXX='vROHR',dropColList=['NAME_i_2L','NAME_k_2L
↳ ']))
    BESCHREIBUNG          IDREFERENZ BAUJAHR HAL IPLANUNG KENNUNG      L LZU   RAU
↳ ZAUS ZEIN ZUML JLAMBS LAMBD A0 ASOLL INDSCHALL KVR AUSFALLZEIT DA   DI   DN KT PN
↳ REHABILITATION REPARATUR  S WSTEIG WTIEFE LTGR_NAME LTGR_BESCHREIBUNG SICHTBARKEIT
↳ VERLEGEART DTRO_NAME          DTRO_BESCHREIBUNG      E fkSTRASSE
↳ fkSRAT          pk                      tk IRTRENN LECKSTART LECKEND
↳ LECKMENGE LECKORT LECKSTATUS QSVB ZVLIMPTNZ KANTENZV      CONT CONT_ID CONT_LFDNR
↳ NAME_i KVR_i TM_i XKOR_i YKOR_i ZKOR_i NAME_k KVR_k TM_k XKOR_k YKOR_k ZKOR_k
↳ pXCor_i pYCor_i pXCor_k pYCor_k          pXCors          pYCors          pWAYPXCors
↳ pWAYPYCors          WAYP
0          NaN  3S4737064599036143765    2017  0          1          0  10000  0  0.25
↳ 0  0  0          1  0.025  1000          0  0          0  0  250  250  0  0
↳          0          0  0          0          0  STDROHR          NaN          1
↳ 999999  STDROHR  Standard-Druckrohre mit di = DN (DIN 2402)  2.1E+11          -1
↳ -1  4737064599036143765  4737064599036143765          0          0          0          0
↳          0          0          0          0  OneLPipe    1001          NaN      I      0
↳ 10  300  600          10      K      0  10  800  600          10          0.0          0.0
↳ 500.0          0.0 [0.0, 500.0] [0.0, 0.0] [0.0, 500.0] [0.0, 0.0] [(300.0, 600.0),
↳ (800.0, 600.0)]
>>> # ---

```

(continues on next page)

(continued from previous page)

```

>>> # Clean Up
>>> # ---
>>> xm.delFiles()
>>> # ---
>>> # LocalHeatingNetwork
>>> # ---
>>> xmlFile=os.path.join(os.path.join(path,testDir),'LocalHeatingNetwork.XML')
>>> xm=Xm(xmlFile=xmlFile)
>>> # ---
>>> # vKNOT
>>> # ---
>>> print(xm._getvXXXXAsOneString(vXXXX='vKNOT',dropColList=['LFKT_NAME','LF','LF_min
↳','LF_max','PVAR_NAME','PH','PH_min','PH_max','PZON_NAME','FSTF_NAME','STOF_NAME',
↳'GMIX_NAME','UTMP_NAME','2L_NAME','2L_KVR','fkHYDR','fkFQPS']))
      NAME                                BESCHREIBUNG IDREFERENZ
↳      CONT CONT_ID CONT_LFDNR CONT_VKNO  KTYP LFAKT QM_EIN QVAR_NAME  QM  QM_
↳min  QM_max KVR  TE  TM      XKOR      YKOR ZKOR                                pk
↳ tk    pXCor  pYCor
0      R-K004                                None                -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu    1001                -1      NaN QKON      1      0      NaN NaN
↳NaN      NaN      2  10  60  2541539  5706361  20  4638663808856251977
↳4638663808856251977    799.0  152.0
1      V-K002                                None                -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu    1001                -1      NaN QKON      1      0      NaN NaN
↳NaN      NaN      1  10  90  2541059  5706265  20  4731792362611615619
↳4731792362611615619    319.0   56.0
2      V-K001                                None                -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu    1001                -1      NaN QKON      1      0      NaN NaN
↳NaN      NaN      1  10  90  2540867  5706228  20  4756962427318766791
↳4756962427318766791    127.0   19.0
3      V-K000                                None                -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu    1001                -1      NaN QKON      1      0      NaN NaN
↳NaN      NaN      1  10  90  2540793  5706209  20  4766681917240867943
↳4766681917240867943    53.0    0.0
4      R-K001                                None                -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu    1001                -1      NaN QKON      1      0      NaN NaN
↳NaN      NaN      2  10  60  2540867  5706228  20  4807712987325933680
↳4807712987325933680    127.0   19.0
5      R-K003                                None                -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu    1001                -1      NaN QKON      1      0      NaN NaN
↳NaN      NaN      2  10  60  2541457  5706345  20  4891048046264179170
↳4891048046264179170    717.0  136.0
6      R-K000                                None                -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu    1001                -1      NaN QKON      1      0      NaN NaN
↳NaN      NaN      2  10  60  2540793  5706209  20  4979785838440534851
↳4979785838440534851    53.0    0.0
7      R-K005                                None                -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu    1001                -1      NaN QKON      1      0      NaN NaN
↳NaN      NaN      2  10  60  2541627  5706363  20  5183147862966701025
↳5183147862966701025    887.0  154.0
8      R-L                                    None                -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu    1001                -1      BHKW QKON      1      0      NaN NaN
↳NaN      NaN      2  10  60  2540740  5706225  20  5356267303828212700
↳5356267303828212700     0.0   16.0
9      R-K002                                None                -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu    1001                -1      NaN QKON      1      0      NaN NaN
↳NaN      NaN      2  10  60  2541059  5706265  20  5364712333175450942
↳5364712333175450942    319.0   56.0

```

(continues on next page)

(continued from previous page)

```

10      V-K004      None      -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu      1001      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      1      10      90      2541539      5706361      20      5370423799772591808
↳5370423799772591808      799.0      152.0
11      V-K005      None      -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu      1001      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      1      10      90      2541627      5706363      20      5444644492819213978
↳5444644492819213978      887.0      154.0
12      R-K007      None      -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu      1001      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      2      10      60      2541899      5706325      20      5508992300317633799
↳5508992300317633799      1159.0      116.0
13      V-K006      None      -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu      1001      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      1      10      90      2541790      5706338      20      5515313800585145571
↳5515313800585145571      1050.0      129.0
14      R-K006      None      -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu      1001      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      2      10      60      2541790      5706338      20      5543326527366090679
↳5543326527366090679      1050.0      129.0
15      V-K003      None      -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu      1001      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      1      10      90      2541457      5706345      20      5646671866542823796
↳5646671866542823796      717.0      136.0
16      V-L      None      -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu      1001      -1      BHKW  QKON      1      0      NaN NaN
↳NaN      NaN      1      10      90      2540740      5706240      20      5736262931552588702
↳5736262931552588702      0.0      31.0
17      V-K007      None      -1  Nahwärmenetz mit 1000 kW
↳Anschlussleistu      1001      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      1      10      90      2541899      5706325      20      5741235692335544560
↳5741235692335544560      1159.0      116.0
18      R2      None      -1
↳      BHKW      1002      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      2      10      60      170      20      20      5002109894154139899
↳5002109894154139899      170.0      20.0
19      V-1      None      -1
↳      BHKW      1002      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      1      10      90      140      160      20      5049461676240771430
↳5049461676240771430      140.0      160.0
20      R3      None      -1
↳      BHKW      1002      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      2      10      60      140      20      20      5219230031772497417
↳5219230031772497417      140.0      20.0
21      PKON-Knoten Druckhaltung - 2 bar Ruhedruck      -1
↳      BHKW      1002      -1      NaN  PKON      1      0      NaN NaN
↳NaN      NaN      2      60      60      200      40      20      5397990465339071638
↳5397990465339071638      200.0      40.0
22      R-1      Anbindung Druckhaltung      -1
↳      BHKW      1002      -1      NaN  QKON      1      0      NaN NaN
↳NaN      NaN      2      10      60      195      20      20      5557222628687032084
↳5557222628687032084      195.0      20.0
>>> # ---
>>> # vROHR
>>> # ---
>>> print (xm._getvXXXXAsOneString(vXXX='vROHR',dropColList=['NAME_i_2L','NAME_k_2L
↳'])

```

(continues on next page)

(continued from previous page)

BESCHREIBUNG	IDREFERENZ	BAUJAHR	HAL	IPLANUNG	KENNUNG	L	LZU	RAU	ZAUS	ZEIN	
→ZUML JLAMBS LAMBD00 ASOLL	INDSCHALL	KVR	AUSFALLZEIT	DA	DI	DN	KT	PN			
→REHABILITATION REPARATUR	S	WSTEIG	WTIEFE	LTGR_NAME		LTGR_BESCHREIBUNG					
→SICHTBARKEIT	VERLEGEART	DTRO_NAME		DTRO_BESCHREIBUNG		E					
→fkSTRASSE	fkSRAT		pk	tk	IRTRENN	LECKSTART					
→LECKEND	LECKMENGE	LECKORT	LECKSTATUS	QSVB	ZVLIMPTNZ	KANTENZV					
→	CONT	CONT_ID	CONT_LFDNR	NAME_i	KVR_i	TM_i	XKOR_i	YKOR_i	ZKOR_i		
→NAME_k	KVR_k	TM_k	XKOR_k	YKOR_k	ZKOR_k	pXCor_i	pYCor_i	pXCor_k	pYCor_k		
→	pXCors		pYCors			pWAYPXCors					
→			pWAYPYCors								
→				WAYP							
0	None	-1	None	0	1	0	88.02	0	0.1	0	0
→0	1	0.025	1000	0	2	NaN	114.3	107.1	100	0.325	NaN
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe			
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11				
→	-1	-1	4613782368750024999	4613782368750024999	0	0	0				
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW				
→Anschlussleistu	1001	-1	R-K004	2	60	2541539	5706361	20	R-		
→K005	2	60	2541627	5706363	20	799.0	152.0	887.0	154.0		
→	[799.0, 887.0]	[152.0, 154.0]	[807.8999999999069, 895.9500000001863]	[140.09999999962747, 142.04999999981374]							
→	[(2541547.9, 5706349.1), (2541635.95, 5706351.05)]										
1	None	-1	None	0	1	0	405.96	0	0.1	0	0
→0	1	0.025	1000	0	1	NaN	114.3	107.1	100	0.325	NaN
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe			
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11				
→	-1	-1	4614949065966596185	4614949065966596185	0	0	0				
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW				
→Anschlussleistu	1001	-1	V-K002	1	90	2541059	5706265	20	V-		
→K003	1	90	2541457	5706345	20	319.0	56.0	717.0	136.0		
→	[319.0, 717.0]	[56.0, 136.0]	[319.0, 716.9500000001863]								
→	[56.049999999813735, 136.0]										
→	[(2541059.0, 5706265.05), (2541456.95, 5706345.0)]										
2	None	-1	None	0	1	0	83.55	0	0.1	0	0
→0	1	0.025	1000	0	2	NaN	114.3	107.1	100	0.325	NaN
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe			
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11				
→	-1	-1	4637102239750163477	4637102239750163477	0	0	0				
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW				
→Anschlussleistu	1001	-1	R-K003	2	60	2541457	5706345	20	R-		
→K004	2	60	2541539	5706361	20	717.0	136.0	799.0	152.0		
→	[717.0, 799.0]	[136.0, 152.0]	[725.8500000000931, 807.8999999999069]	[124.04999999981374, 140.09999999962747]							
→	[(2541465.85, 5706333.05), (2541547.9, 5706349.1)]										
3	None	-1	None	0	1	0	88.02	0	0.1	0	0
→0	1	0.025	1000	0	1	NaN	114.3	107.1	100	0.325	NaN
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe			
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11				
→	-1	-1	4713733238627697042	4713733238627697042	0	0	0				
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW				
→Anschlussleistu	1001	-1	V-K004	1	90	2541539	5706361	20	V-		
→K005	1	90	2541627	5706363	20	799.0	152.0	887.0	154.0		
→	[799.0, 887.0]	[152.0, 154.0]	[799.0, 887.04999999998137]								
→	[152.0, 154.0]										
→	[(2541539.0, 5706361.0), (2541627.05, 5706363.0)]										
4	None	-1	None	0	1	0	195.53	0	0.1	0	0
→0	1	0.025	1000	0	1	NaN	114.3	107.1	100	0.325	NaN
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe			
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11				
→	-1	-1	4789218195240364437	4789218195240364437	0	0	0				
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW				
→Anschlussleistu	1001	-1	V-K001	1	90	2540867	5706228	20	V-		
→K002	1	90	2541059	5706265	20	127.0	19.0	319.0	56.0		
→	[127.0, 319.0]	[19.0, 56.0]	[127.0, 319.0]								

(continues on next page)

(continued from previous page)

```

5      None      -1      None      0      1      0 109.77  0 0.1  0  0
→0      1  0.025  1000      0  2      NaN 114.3 107.1 100 0.325 NaN
→      NaN      NaN  3.6      NaN      NaN KUMANRO Beschreibung Leitungsgruppe
→      1  999999 KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1 -1 4945727430885351042 4945727430885351042 0 0 0
→      0  0  0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001 -1 R-K006 2 60 2541790 5706338 20 R-
→K007 2 60 2541899 5706325 20 1050.0 129.0 1159.0 116.0 [1050.
→0, 1159.0] [129.0, 116.0] [1058.8500000000931, 1167.8999999999069]
→[117.0, 104.099999999962747]
→[(2541798.85, 5706326.0), (2541907.9, 5706313.1)]
6      None      -1      None      0      1      0 76.4  0 0.1  0  0
→0      1  0.025  1000      0  1      NaN 114.3 107.1 100 0.325 NaN
→      NaN      NaN  3.6      NaN      NaN KUMANRO Beschreibung Leitungsgruppe
→      1  999999 KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1 -1 4984202422877610920 4984202422877610920 0 0 0
→      0  0  0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001 -1 V-K000 1 90 2540793 5706209 20 V-
→K001 1 90 2540867 5706228 20 53.0 0.0 127.0 19.0
→[53.0, 127.0] [0.0, 19.0] [53.0499999999813735, 127.0]
→[-0.0499999999813735485, 19.0]
→[(2540793.05, 5706208.95), (2540867.0, 5706228.0)]
7      None      -1      None      0      1      0 83.55  0 0.1  0  0
→0      1  0.025  1000      0  1      NaN 114.3 107.1 100 0.325 NaN
→      NaN      NaN  3.6      NaN      NaN KUMANRO Beschreibung Leitungsgruppe
→      1  999999 KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1 -1 5037777106796980248 5037777106796980248 0 0 0
→      0  0  0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001 -1 V-K003 1 90 2541457 5706345 20 V-
→K004 1 90 2541539 5706361 20 717.0 136.0 799.0 152.0
→[717.0, 799.0] [136.0, 152.0] [716.9500000001863, 799.0]
→[136.0, 152.0]
→[(2541456.95, 5706345.0), (2541539.0, 5706361.0)]
8      None      -1      None      0      1      0 164.91  0 0.1  0  0
→0      1  0.025  1000      0  1      NaN 114.3 107.1 100 0.325 NaN
→      NaN      NaN  3.6      NaN      NaN KUMANRO Beschreibung Leitungsgruppe
→      1  999999 KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1 -1 5123819811204259837 5123819811204259837 0 0 0
→      0  0  0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001 -1 V-K005 1 90 2541627 5706363 20 V-
→K006 1 90 2541790 5706338 20 887.0 154.0 1050.0 129.0 [887.
→0, 1050.0] [154.0, 129.0] [887.04999999998137, 1049.9500000001863]
→[154.0, 128.95000000018626]
→[(2541627.05, 5706363.0), (2541789.95, 5706337.95)]
9      None      -1      None      0      1      0 195.53  0 0.1  0  0
→0      1  0.025  1000      0  2      NaN 114.3 107.1 100 0.325 NaN
→      NaN      NaN  3.6      NaN      NaN KUMANRO Beschreibung Leitungsgruppe
→      1  999999 KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1 -1 5266224553324203132 5266224553324203132 0 0 0
→      0  0  0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001 -1 R-K001 2 60 2540867 5706228 20 R-
→K002 2 60 2541059 5706265 20 127.0 19.0 319.0 56.0
→[127.0, 319.0] [19.0, 56.0] [135.899999999990687, 327.899999999990687] [7.
→04999999998137355, 44.099999999962747]
→[(2540875.9, 5706216.05), (2541067.9, 5706253.1)]
10     None      -1      None      0      1      0 405.96  0 0.1  0  0
→0      1  0.025  1000      0  2      NaN 114.3 107.1 100 0.325 NaN
→      NaN      NaN  3.6      NaN      NaN KUMANRO Beschreibung Leitungsgruppe
→      1  999999 KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1 -1 5379365049009065623 5379365049009065623 0 0 0
→      0  0  0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001 -1 R-K002 2 60 2541059 5706265 20 R-
→K003 2 60 2541457 5706345 20 319.0 56.0 717.0 136.0
→[319.0, 717.0] [56.0, 136.0] [327.899999999990687, 725.8500000000931] [44.
→04999999998137355, 44.099999999962747]

```

(continues on next page)

(continued from previous page)

```

11      None      -1      None      0      1      0 164.91  0 0.1  0  0
→0      1  0.025  1000      0  2      NaN 114.3 107.1 100 0.325 NaN
→      NaN      NaN  3.6      NaN      NaN      KUMANRO Beschreibung Leitungsgruppe
→      1      999999      KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1      -1  5611703699850694889 5611703699850694889      0      0      0
→      0      0      0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001      -1 R-K005      2  60 2541627 5706363      20 R-
→K006      2  60 2541790 5706338      20 887.0 154.0 1050.0 129.0 [887.
→0, 1050.0] [154.0, 129.0] [895.9500000001863, 1058.8500000000931]
→[142.04999999981374, 117.0]
→[(2541635.95, 5706351.05), (2541798.85, 5706326.0)]
12      None      -1      None      0      1      0 109.77  0 0.1  0  0
→0      1  0.025  1000      0  1      NaN 114.3 107.1 100 0.325 NaN
→      NaN      NaN  3.6      NaN      NaN      KUMANRO Beschreibung Leitungsgruppe
→      1      999999      KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1      -1  5620197984230756681 5620197984230756681      0      0      0
→      0      0      0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001      -1 V-K006      1  90 2541790 5706338      20 V-
→K007      1  90 2541899 5706325      20 1050.0 129.0 1159.0 116.0 [1050.
→0, 1159.0] [129.0, 116.0] [1049.9500000001863, 1159.0] [128.
→95000000018626, 116.04999999981374]
→[(2541789.95, 5706337.95), (2541899.0, 5706325.05)]
13      None      -1      None      0      1      0 76.4  0 0.1  0  0
→0      1  0.025  1000      0  2      NaN 114.3 107.1 100 0.325 NaN
→      NaN      NaN  3.6      NaN      NaN      KUMANRO Beschreibung Leitungsgruppe
→      1      999999      KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1      -1  5647213228462830353 5647213228462830353      0      0      0
→      0      0      0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001      -1 R-K000      2  60 2540793 5706209      20 R-
→K001      2  60 2540867 5706228      20 53.0 0.0 127.0 19.0
→[53.0, 127.0] [0.0, 19.0] [61.950000000186265, 135.89999999990687]
→[-12.0, 7.0499999998137355]
→[(2540801.95, 5706197.0), (2540875.9, 5706216.05)]
14      None      -1      None      0      1      0 73.42  0 0.1  0  0
→0      1  0.025  1000      0  2      NaN 168.3 160.3 150 0.45 NaN
→      NaN      NaN  4      NaN      NaN      KUMANRO Beschreibung Leitungsgruppe
→      1      999999      KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1      -1  4769996343148550485 4769996343148550485      0      0      0
→      0      0      0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001      -1 R-L      2  60 2540740 5706225      20 R-
→K000      2  60 2540793 5706209      20 0.0 16.0 53.0 0.0
→[0.0, 53.0] [16.0, 0.0] [0.0, 24.0, 45.0, 61.950000000186265]
→[16.0, 16.0, -12.0, -12.0] [(2540740.0, 5706225.0), (2540764.0, 5706225.0),
→(2540785.0, 5706197.0), (2540801.95, 5706197.0)]
15      None      -1      None      0      1      0 68.6  0 0.1  0  0
→0      1  0.025  1000      0  1      NaN 168.3 160.3 150 0.45 NaN
→      NaN      NaN  4      NaN      NaN      KUMANRO Beschreibung Leitungsgruppe
→      1      999999      KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT 2.1E+11
→ -1      -1  4939422678063487923 4939422678063487923      0      0      0
→      0      0      0  0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu 1001      -1 V-L      1  90 2540740 5706240      20 V-
→K000      1  90 2540793 5706209      20 0.0 31.0 53.0 0.0
→[0.0, 53.0] [31.0, 0.0] [0.0, 30.0, 53.049999999813735] [31.0,
→31.0, -0.049999999813735485] [(2540740.0, 5706240.0),
→(2540770.0, 5706240.0), (2540793.05, 5706208.95)]
>>> # ---
>>> # vWBLZ

```

(continues on next page)

(continued from previous page)

```

>>> # ---
>>> print("'{:s}'".format(repr(xm.dataFrames['vWBLZ']).replace('\n', '\n  ')))
'''   AKTIV BESCHREIBUNG IDIM          NAME OBJTYPE          OBJID
-> pk
  0      1   Wärmebilanz      0      BLNZ1      KNOT  4731792362611615619
->5579937562601803472
  1      1   Wärmebilanz      0      BLNZ1      KNOT  5364712333175450942
->5579937562601803472
  2      1   Wärmebilanz      0      BLNZ1u5      KNOT  5183147862966701025
->5187647097142898375
  3      1   Wärmebilanz      0      BLNZ1u5      KNOT  5444644492819213978
->5187647097142898375
  4      1   Wärmebilanz      0      BLNZ1u5      KNOT  4731792362611615619
->5187647097142898375
  5      1   Wärmebilanz      0      BLNZ1u5      KNOT  5364712333175450942
->5187647097142898375
  6      1   Wärmebilanz      0      BLNZ1u5u7      KNOT  5183147862966701025
->4694700216019268978
  7      1   Wärmebilanz      0      BLNZ1u5u7      KNOT  5444644492819213978
->4694700216019268978
  8      1   Wärmebilanz      0      BLNZ1u5u7      KNOT  4731792362611615619
->4694700216019268978
  9      1   Wärmebilanz      0      BLNZ1u5u7      KNOT  5364712333175450942
->4694700216019268978
 10      1   Wärmebilanz      0      BLNZ1u5u7      KNOT  5508992300317633799
->4694700216019268978
 11      1   Wärmebilanz      0      BLNZ1u5u7      KNOT  5741235692335544560
->4694700216019268978
 12      1   Wärmebilanz      0      BLNZ5      KNOT  5183147862966701025
->5581152085151655438
 13      1   Wärmebilanz      0      BLNZ5      KNOT  5444644492819213978
->5581152085151655438'''
>>> # ---
>>> # vAGSN
>>> # ---
>>> print(xm._getvXXXXAsOneString(vXXXX='vAGSN',end=7,dropColList=['nrObjIdTypeInAgsn
->','compNr']))
      LFDNR          NAME AKTIV OBJTYPE          OBJID
->          pk          tk nrObjIdInAgsn Layer nextNODE
0      1   Netzdruckdiagramm VL/RL: BHKW - Netzende  101   ROHR  4939422678063487923
->5252525269080005909  5252525269080005909          1      1   V-K000
1      1   Netzdruckdiagramm VL/RL: BHKW - Netzende  101   ROHR  4984202422877610920
->5252525269080005909  5252525269080005909          2      1   V-K001
2      1   Netzdruckdiagramm VL/RL: BHKW - Netzende  101   ROHR  4789218195240364437
->5252525269080005909  5252525269080005909          3      1   V-K002
3      1   Netzdruckdiagramm VL/RL: BHKW - Netzende  101   ROHR  4614949065966596185
->5252525269080005909  5252525269080005909          4      1   V-K003
4      1   Netzdruckdiagramm VL/RL: BHKW - Netzende  101   ROHR  5037777106796980248
->5252525269080005909  5252525269080005909          5      1   V-K004
5      1   Netzdruckdiagramm VL/RL: BHKW - Netzende  101   ROHR  4713733238627697042
->5252525269080005909  5252525269080005909          6      1   V-K005
6      1   Netzdruckdiagramm VL/RL: BHKW - Netzende  101   ROHR  5123819811204259837
->5252525269080005909  5252525269080005909          7      1   V-K006
>>> # ---
>>> # vFWVB
>>> # ---
>>> print("'{:s}'".format(repr(xm.dataFrames['vFWVB']).replace('\n', '\n  ')))

```

(continues on next page)

(continued from previous page)

```

'''  BESCHREIBUNG IDREFERENZ  WO  LFK  WOLFK  TVL0  TRS0  LFKT      W  W_min  W_max
↳INDTR  TRSK  VTYP  DPHAUS  IMBG  IRFV      pk      tk
↳NAME_i  KVR_i  TM_i  XKOR_i  YKOR_i  ZKOR_i  pXCor_i  pYCor_i  NAME_k  KVR_k  TM_k
↳XKOR_k  YKOR_k  ZKOR_k  pXCor_k  pYCor_k      CONT
↳CONT_ID  CONT_LFDNR      WBLZ
  0      1      -1  200  0.8  160.0  90  50  LFKT  160.0  160.0  160.0
↳  1      55      14      0.7      0  0.0  4643800032883366034  4643800032883366034  V-
↳K002      1      90  2541059  5706265      20  319.0      56.0  R-K002      2  60
↳2541059  5706265      20  319.0      56.0  Nahwärmenetz mit 1000 kW Anschlussleistu
↳  1001      -1  [BLNZ1, BLNZ1u5, BLNZ1u5u7]
  1      3      -1  200  1.0  200.0  90  65  LFKT  200.0  200.0  200.0
↳  1      65      14      0.7      0  0.0  4704603947372595298  4704603947372595298  V-
↳K004      1      90  2541539  5706361      20  799.0      152.0  R-K004      2  60
↳2541539  5706361      20  799.0      152.0  Nahwärmenetz mit 1000 kW Anschlussleistu
↳  1001      -1  []
  2      4      -1  200  0.8  160.0  90  60  LFKT  160.0  160.0  160.0
↳  1      60      14      0.7      0  0.0  5121101823283893406  5121101823283893406  V-
↳K005      1      90  2541627  5706363      20  887.0      154.0  R-K005      2  60
↳2541627  5706363      20  887.0      154.0  Nahwärmenetz mit 1000 kW Anschlussleistu
↳  1001      -1  [BLNZ1u5, BLNZ1u5u7, BLNZ5]
  3      5      -1  200  0.8  160.0  90  55  LFKT  160.0  160.0  160.0
↳  1      55      14      0.7      0  0.0  5400405917816384862  5400405917816384862  V-
↳K007      1      90  2541899  5706325      20  1159.0      116.0  R-K007      2  60
↳2541899  5706325      20  1159.0      116.0  Nahwärmenetz mit 1000 kW Anschlussleistu
↳  1001      -1  [BLNZ1u5u7]
  4      2      -1  200  0.6  120.0  90  60  LFKT  120.0  120.0  120.0
↳  1      62      14      0.7      0  0.0  5695730293103267172  5695730293103267172  V-
↳K003      1      90  2541457  5706345      20  717.0      136.0  R-K003      2  60
↳2541457  5706345      20  717.0      136.0  Nahwärmenetz mit 1000 kW Anschlussleistu
↳  1001      -1  []'''
>>> # ---
>>> # vLAYR
>>> # ---
>>> # print("'{s}':".format(repr(xm.dataFrames['vLAYR']).sort_values(['LFDNR', 'NAME
↳', 'OBJTYPE', 'OBJID'], ascending=True)).replace('\n', '\n ')))

```

```

>>> vStr=xm.getVersion(type='BZ')
>>> import re
>>> m=re.search('Sir(?P<Db3s>[DBdb3Ss]{2})-(?P<Major>\d+)-(?P<Minor>\d+)$',vStr) # i.
↳e. Sir3S-90-10
>>> minorVer=int(m.group('Minor'))
>>> vLAYR=xm.dataFrames['vLAYR'].copy()
>>> #print(vLAYR.shape)
>>> if minorVer>=12:
...   vLAYR['OBJTYPE']=vLAYR['OBJTYPE'].map(lambda x: 'TEXT' if x=='GTXT' else x)
...   vLAYR['OBJTYPE']=vLAYR['OBJTYPE'].map(lambda x: 'BSYM' if x=='CONT' else x)
...   sortList=['LFDNR', 'NAME', 'OBJTYPE', 'OBJID']
... else:
...   sortList=None
>>> # Analyse der Unterschiede zu Testzwecken ...
>>> sGot=xm._getvXXXXAsOneString(vXXXX='vLAYR', sortList=sortList, index=False)
>>> sExp='''LFDNR      NAME  OBJTYPE      OBJID      pk
↳      tk  nrObjInGroup  nrObjtypeInGroup  1      Vorlauf      FWES
↳5638756766880678918  5206516471428693478  5206516471428693478      1
↳      1      1      Vorlauf      KNOT  4731792362611615619  5206516471428693478
↳5206516471428693478      1      1      1      Vorlauf      KNOT
↳4756962427318766791  5206516471428693478  5206516471428693478      1
↳      2      1      Vorlauf      KNOT  4766681917240867943  5206516471428693478
↳5206516471428693478      1      3      1      Vorlauf      KNOT
465049461676240771430  5206516471428693478  5206516471428693478      1
↳      4      1      Vorlauf      KNOT  5370423799772591808  5206516471428693478
↳5206516471428693478      1      5      1      Vorlauf      KNOT
↳5444644492819213978  5206516471428693478  5206516471428693478      1
↳      6      1      Vorlauf      KNOT  5515313800585145571  5206516471428693478

```

(continues on next page)

(continued from previous page)

```
>>> import difflib
>>> s = difflib.SequenceMatcher(None, sExp, sGot)
>>> for block in s.get_matching_blocks():
...     pass
...     # print(block)
>>> print(xm._getvXXXXAsOneString(vXXXX='vLAYR', sortList=sortList, index=False))
```

LFDNR	NAME	OBJTYPE	OBJID	pk
→ tk nrObjInGroup nrObjtypeInGroup				
1	Vorlauf	FWES	5638756766880678918	5206516471428693478
→ 5206516471428693478			1	1
1	Vorlauf	KNOT	4731792362611615619	5206516471428693478
→ 5206516471428693478			1	1
1	Vorlauf	KNOT	4756962427318766791	5206516471428693478
→ 5206516471428693478			1	2
1	Vorlauf	KNOT	4766681917240867943	5206516471428693478
→ 5206516471428693478			1	3
1	Vorlauf	KNOT	5049461676240771430	5206516471428693478
→ 5206516471428693478			1	4
1	Vorlauf	KNOT	5370423799772591808	5206516471428693478
→ 5206516471428693478			1	5
1	Vorlauf	KNOT	5444644492819213978	5206516471428693478
→ 5206516471428693478			1	6
1	Vorlauf	KNOT	5515313800585145571	5206516471428693478
→ 5206516471428693478			1	7
1	Vorlauf	KNOT	5646671866542823796	5206516471428693478
→ 5206516471428693478			1	8
1	Vorlauf	KNOT	5736262931552588702	5206516471428693478
→ 5206516471428693478			1	9
1	Vorlauf	KNOT	5741235692335544560	5206516471428693478
→ 5206516471428693478			1	10
1	Vorlauf	ROHR	4614949065966596185	5206516471428693478
→ 5206516471428693478			1	1
1	Vorlauf	ROHR	4713733238627697042	5206516471428693478
→ 5206516471428693478			1	2
1	Vorlauf	ROHR	4789218195240364437	5206516471428693478
→ 5206516471428693478			1	3
1	Vorlauf	ROHR	4939422678063487923	5206516471428693478
→ 5206516471428693478			1	4
1	Vorlauf	ROHR	4984202422877610920	5206516471428693478
→ 5206516471428693478			1	5
1	Vorlauf	ROHR	5037777106796980248	5206516471428693478
→ 5206516471428693478			1	6
1	Vorlauf	ROHR	5123819811204259837	5206516471428693478
→ 5206516471428693478			1	7
1	Vorlauf	ROHR	5620197984230756681	5206516471428693478
→ 5206516471428693478			1	8
1	Vorlauf	VENT	4678923650983295610	5206516471428693478
→ 5206516471428693478			1	1
2	Rücklauf	KLAP	4801110583764519435	4693347477612662930
→ 4693347477612662930			1	1
2	Rücklauf	KNOT	4638663808856251977	4693347477612662930
→ 4693347477612662930			1	1
2	Rücklauf	KNOT	4807712987325933680	4693347477612662930
→ 4693347477612662930			1	2
2	Rücklauf	KNOT	4891048046264179170	4693347477612662930
→ 4693347477612662930			1	3
2	Rücklauf	KNOT	4979785838440534851	4693347477612662930
→ 4693347477612662930			1	4

(continues on next page)

(continued from previous page)

2	Rücklauf	KNOT	5002109894154139899	4693347477612662930	└
↪4693347477612662930			1	5	
2	Rücklauf	KNOT	5183147862966701025	4693347477612662930	└
↪4693347477612662930			1	6	
2	Rücklauf	KNOT	5219230031772497417	4693347477612662930	└
↪4693347477612662930			1	7	
2	Rücklauf	KNOT	5356267303828212700	4693347477612662930	└
↪4693347477612662930			1	8	
2	Rücklauf	KNOT	5364712333175450942	4693347477612662930	└
↪4693347477612662930			1	9	
2	Rücklauf	KNOT	5397990465339071638	4693347477612662930	└
↪4693347477612662930			1	10	
2	Rücklauf	KNOT	5508992300317633799	4693347477612662930	└
↪4693347477612662930			1	11	
2	Rücklauf	KNOT	5543326527366090679	4693347477612662930	└
↪4693347477612662930			1	12	
2	Rücklauf	KNOT	5557222628687032084	4693347477612662930	└
↪4693347477612662930			1	13	
2	Rücklauf	PUMP	5481331875203087055	4693347477612662930	└
↪4693347477612662930			1	1	
2	Rücklauf	ROHR	4613782368750024999	4693347477612662930	└
↪4693347477612662930			1	1	
2	Rücklauf	ROHR	4637102239750163477	4693347477612662930	└
↪4693347477612662930			1	2	
2	Rücklauf	ROHR	4769996343148550485	4693347477612662930	└
↪4693347477612662930			1	3	
2	Rücklauf	ROHR	4945727430885351042	4693347477612662930	└
↪4693347477612662930			1	4	
2	Rücklauf	ROHR	5266224553324203132	4693347477612662930	└
↪4693347477612662930			1	5	
2	Rücklauf	ROHR	5379365049009065623	4693347477612662930	└
↪4693347477612662930			1	6	
2	Rücklauf	ROHR	5611703699850694889	4693347477612662930	└
↪4693347477612662930			1	7	
2	Rücklauf	ROHR	5647213228462830353	4693347477612662930	└
↪4693347477612662930			1	8	
2	Rücklauf	VENT	4897018421024717974	4693347477612662930	└
↪4693347477612662930			1	1	
2	Rücklauf	VENT	5525310316015533093	4693347477612662930	└
↪4693347477612662930			1	2	
3	Kundenanlagen	FWVB	4643800032883366034	5003333277973347346	└
↪5003333277973347346			1	1	
3	Kundenanlagen	FWVB	4704603947372595298	5003333277973347346	└
↪5003333277973347346			1	2	
3	Kundenanlagen	FWVB	5121101823283893406	5003333277973347346	└
↪5003333277973347346			1	3	
3	Kundenanlagen	FWVB	5400405917816384862	5003333277973347346	└
↪5003333277973347346			1	4	
3	Kundenanlagen	FWVB	5695730293103267172	5003333277973347346	└
↪5003333277973347346			1	5	
4	BHKW	BSYM	5043395081363401573	5555393404073362943	└
↪5555393404073362943			1	1	
4	BHKW	TEXT	5056836766824229789	5555393404073362943	└
↪5555393404073362943			1	1	
4	BHKW	TEXT	5329748935118523443	5555393404073362943	└
↪5555393404073362943			1	2	
5	Texte	ARRW	4664845735864571219	5394410243594912680	└
↪5394410243594912680			1	1	

(continues on next page)

(continued from previous page)

5	Texte	ARRW	4902474974831811106	5394410243594912680	1
↪5394410243594912680			1	2	
5	Texte	ARRW	5026846801782366678	5394410243594912680	1
↪5394410243594912680			1	3	
5	Texte	ARRW	5688313372729413840	5394410243594912680	1
↪5394410243594912680			1	4	
5	Texte	NRCV	4681213816714574464	5394410243594912680	1
↪5394410243594912680			1	1	
5	Texte	NRCV	4857294696992797631	5394410243594912680	1
↪5394410243594912680			1	2	
5	Texte	NRCV	4914949875368816179	5394410243594912680	1
↪5394410243594912680			1	3	
5	Texte	NRCV	4946584950744559030	5394410243594912680	1
↪5394410243594912680			1	4	
5	Texte	NRCV	4968703141722117357	5394410243594912680	1
↪5394410243594912680			1	5	
5	Texte	NRCV	5091374651838464239	5394410243594912680	1
↪5394410243594912680			1	6	
5	Texte	NRCV	5097127385155151127	5394410243594912680	1
↪5394410243594912680			1	7	
5	Texte	NRCV	5179988968597313889	5394410243594912680	1
↪5394410243594912680			1	8	
5	Texte	NRCV	5281885868749421521	5394410243594912680	1
↪5394410243594912680			1	9	
5	Texte	NRCV	5410904806390050339	5394410243594912680	1
↪5394410243594912680			1	10	
5	Texte	NRCV	5476262878682325254	5394410243594912680	1
↪5394410243594912680			1	11	
5	Texte	NRCV	5557806245003742769	5394410243594912680	1
↪5394410243594912680			1	12	
5	Texte	RECT	4994817837124479818	5394410243594912680	1
↪5394410243594912680			1	1	
5	Texte	RPFL	5158870568935841216	5394410243594912680	1
↪5394410243594912680			1	1	
5	Texte	TEXT	4628671704393700430	5394410243594912680	1
↪5394410243594912680			1	1	
5	Texte	TEXT	4654104397990769217	5394410243594912680	1
↪5394410243594912680			1	2	
5	Texte	TEXT	4666644549022031339	5394410243594912680	1
↪5394410243594912680			1	3	
5	Texte	TEXT	4693143208412077585	5394410243594912680	1
↪5394410243594912680			1	4	
5	Texte	TEXT	4768731522550494423	5394410243594912680	1
↪5394410243594912680			1	5	
5	Texte	TEXT	4770844990228490264	5394410243594912680	1
↪5394410243594912680			1	6	
5	Texte	TEXT	4782197969172967134	5394410243594912680	1
↪5394410243594912680			1	7	
5	Texte	TEXT	4855692488683645764	5394410243594912680	1
↪5394410243594912680			1	8	
5	Texte	TEXT	4965628942555351751	5394410243594912680	1
↪5394410243594912680			1	9	
5	Texte	TEXT	4995961504641886710	5394410243594912680	1
↪5394410243594912680			1	10	
5	Texte	TEXT	5017907661719368413	5394410243594912680	1
↪5394410243594912680			1	11	
5	Texte	TEXT	5028052147238787802	5394410243594912680	1
↪5394410243594912680			1	12	

(continues on next page)

(continued from previous page)

```

5      Texte      TEXT  5036153631350515544  5394410243594912680  _
↪5394410243594912680      1      13
5      Texte      TEXT  5054433315422452796  5394410243594912680  _
↪5394410243594912680      1      14
5      Texte      TEXT  5108336975548011049  5394410243594912680  _
↪5394410243594912680      1      15
5      Texte      TEXT  5262441422409836340  5394410243594912680  _
↪5394410243594912680      1      16
5      Texte      TEXT  5297832234834839298  5394410243594912680  _
↪5394410243594912680      1      17
5      Texte      TEXT  5370727463979416592  5394410243594912680  _
↪5394410243594912680      1      18
5      Texte      TEXT  5421223289472778073  5394410243594912680  _
↪5394410243594912680      1      19
5      Texte      TEXT  5501963349880613918  5394410243594912680  _
↪5394410243594912680      1      20
5      Texte      TEXT  5502619581048467908  5394410243594912680  _
↪5394410243594912680      1      21
5      Texte      TEXT  5540395812045688781  5394410243594912680  _
↪5394410243594912680      1      22
5      Texte      TEXT  5550982489075668484  5394410243594912680  _
↪5394410243594912680      1      23
5      Texte      TEXT  5610916400841895317  5394410243594912680  _
↪5394410243594912680      1      24
5      Texte      TEXT  5646820849868629537  5394410243594912680  _
↪5394410243594912680      1      25
5      Texte      TEXT  5696590398594231893  5394410243594912680  _
↪5394410243594912680      1      26
5      Texte      TEXT  5697088036451277538  5394410243594912680  _
↪5394410243594912680      1      27
>>>
>>> # ---
>>> # vGTXt
>>> # ---
>>> print(xm._getvXXXXAsOneString(vXXXX='vGTXt', sortList=['CONT_ID', 'pk'], index=False,
↪header=False))
Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1
↪numerische Anzeige:  4614148870174765680  4614148870174765680
↪      (219.0, -278.0)
Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1
↪Georeferenzpunkt 2  4628671704393700430  4628671704393700430  (1115.
↪9500000001863, -323.0)
Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1
↪      Block  4666644549022031339  4666644549022031339  (-
↪58.0, -77.0)
Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1
↪numerische Anzeige  4693143208412077585  4693143208412077585
↪      (1211.0, -9.0)
Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1
↪Knoten und Rohre  4995961504641886710  4995961504641886710
↪      (570.0, -49.0)
Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1
↪Vorlaufstrang  5017907661719368413  5017907661719368413  (358.20699999993667, 220.
↪394999999955297)
Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1
↪LocalHeatingNetwork  5028052147238787802  5028052147238787802
↪      (1163.0, 536.0)

```

(continues on next page)

(continued from previous page)

```

Nahwärmenetz mit 1000 kW Anschlussleistu 1001 -1 Tel. 05131 - 4980-0 ; Fax.
→05131 - 4980-15 5054433315422452796 5054433315422452796 (-
→230.0, -1143.0)
Nahwärmenetz mit 1000 kW Anschlussleistu 1001 -1 Kontrolle: DH-
→Massenstrom 5100960407865990868 5100960407865990868 (-
→60.0, -160.0)
Nahwärmenetz mit 1000 kW Anschlussleistu 1001 -1
→Wärmebilanz: 3 Kunden 5150752151066924202 5150752151066924202
→ (219.0, -318.0)
Nahwärmenetz mit 1000 kW Anschlussleistu 1001 -1 eMail. info@3SConsult.de ; www.
→3SConsult.de 5370727463979416592 5370727463979416592 (-
→230.0, -1204.0)
Nahwärmenetz mit 1000 kW Anschlussleistu 1001 -1 Differenzdruck VL-/
→ RL-Knoten 5502619581048467908 5502619581048467908
→(1211.0, -49.0)
Nahwärmenetz mit 1000 kW Anschlussleistu 1001 -1
→Kundenanlage 5540395812045688781 5540395812045688781 (1131.9500000001863, 283.
→95000000018626)
Nahwärmenetz mit 1000 kW Anschlussleistu 1001 -1
→Fernwärmeverbraucher 5550982489075668484 5550982489075668484
→ (1050.0, 239.0)
Nahwärmenetz mit 1000 kW Anschlussleistu 1001 -1
→Rücklaufstrang 5610916400841895317 5610916400841895317
→ (570.0, -9.0)
Nahwärmenetz mit 1000 kW Anschlussleistu 1001 -1
→Knoten und Rohre 5646820849868629537 5646820849868629537 (358.20699999993667,
→174.39499999955297)
BHKW 1002 -1
→Fernwärmeeinspeiser 4654104397990769217 4654104397990769217
→ (115.0, 80.0)
BHKW 1002 -1
→ Pumpe 4768731522550494423 4768731522550494423
→(175.0, 25.0)
BHKW 1002 -1
→Wärmebilanz Netz: 4770844990228490264 4770844990228490264
→ (90.0, 160.0)
BHKW 1002 -1
→Speicherung 4782197969172967134 4782197969172967134
→(110.0, 140.0)
BHKW 1002 -1
→Richtungspfeil 4855692488683645764 4855692488683645764
→(220.0, 105.0)
BHKW 1002 -1
→ Verluste 4965628942555351751 4965628942555351751
→(110.0, 145.0)
BHKW 1002 -1 (Element
→verbinden) 5036153631350515544 5036153631350515544
→(150.0, 90.0)
BHKW 1002 -1 BHKW Modul
→1000 kW therm. 5056836766824229789 5056836766824229789
→ (35.0, 55.0)
BHKW 1002 -1
→ Ventil 5108336975548011049 5108336975548011049
→(205.0, 25.0)
BHKW 1002 -1
→ Verbrauch 5262441422409836340 5262441422409836340
→(110.0, 150.0)

```

(continues on next page)

(continued from previous page)

```

BHKW 1002 -1
↳Einspeisung 5297832234834839298 5297832234834839298
↳(110.0, 155.0)

BHKW 1002 -1
↳Druckhaltung 2 bar 5329748935118523443 5329748935118523443
↳(180.0, 65.0)

BHKW 1002 -1
↳Numerische Anzeige 5421223289472778073 5421223289472778073
↳(190.0, 115.0)

BHKW 1002 -1
↳Verbindungsline 5501963349880613918 5501963349880613918
↳(150.0, 95.0)

BHKW 1002 -1
↳(Text) 5696590398594231893 5696590398594231893
↳(35.0, 50.0)

BHKW 1002 -1
↳ Klappe 5697088036451277538 5697088036451277538
↳(145.0, 25.0)
>>> # ---
>>> # vNRCV
>>> # ---
>>> print (xm._getvXXXXAsOneString(vXXXX='vNRCV',end=14,dropColList=['DPGR','CONT_LFDNR
↳','pk_ROWS'],sortList=['OBJTYPE'
... , 'fkOBJTYPE' # 90-12 in BZ
... , 'ATTRTYPE', 'cRefLfdNr']))
cRefLfdNr          CONT CONT_ID OBJTYPE
↳fkOBJTYPE ATTRTYPE      tk_ROWS      pk      tk
↳
↳          pXYLB
0          1          BHKW 1002 FWES
↳5638756766880678918 W 5762106696740202356 4857294696992797631
↳4857294696992797631 (90.0, 65.0)
1          1          BHKW 1002 KNOT
↳5049461676240771430 T 4723443975311885965 5097127385155151127
↳5097127385155151127 (90.0, 95.0)
2          1          BHKW 1002 KNOT
↳5219230031772497417 T 5602301870151014230 5557806245003742769
↳5557806245003742769 (90.0, 35.0)
3          1          BHKW 1002 KNOT
↳5356267303828212700 PH 5000989080893535213 4968703141722117357
↳4968703141722117357 (220.0, 25.0)
4          1 Nahwärmenetz mit 1000 kW Anschlussleistu 1001 KNOT
↳5397990465339071638 QM 5134531789044068877 5410059595276504750
↳5410059595276504750 (91.0, -94.0)
5          2          BHKW 1002 KNOT
↳5397990465339071638 QM 5134531789044068877 5357021981944933535
↳5357021981944933535 (184.999999464624, 57.99999953107601)
6          1          BHKW 1002 KNOT
↳5736262931552588702 PH 4754881272083464445 4681213816714574464
↳4681213816714574464 (220.0, 85.0)
7          1 Nahwärmenetz mit 1000 kW Anschlussleistu 1001 KNOT
↳5741235692335544560 DP 4949183695502554728 4914949875368816179
↳4914949875368816179 (1234.0, 83.0)
8          1          BHKW 1002 PUMP
↳5481331875203087055 N 5563842594211689762 5091374651838464239
↳5091374651838464239 (170.0, 45.0)
9          1          BHKW 1002 VENT
↳4678923650983295610 QM 5126307362398248950 5410904806390050339
↳5410904806390050339 (200.0, 110.0)

```

(continues on next page)

(continued from previous page)

```

10      1  Nahwärmenetz mit 1000 kW Anschlussleistu      1001      WBLZ
→4694700216019268978      WVB  4778244458749966216  4991097791264453745
→4991097791264453745      (354.0, -225.0)
11      1      BHKW      1002      WBLZ
→5262603207038486299      WES  5690691957596882133  5179988968597313889
→5179988968597313889      (90.0, 155.0)
12      1      BHKW      1002      WBLZ
→5262603207038486299      WSPEI  5153847813311339683  4946584950744559030
→4946584950744559030      (90.0, 140.0)
13      1      BHKW      1002      WBLZ
→5262603207038486299      WVB  5214984699859365639  5281885868749421521
→5281885868749421521      (90.0, 150.0)
>>> # ---
>>> # MxSync() - without Mx-Object
>>> # ---
>>> vROHR=xm.dataFrames['vROHR']
>>> vROHR.shape
(16, 74)
>>> 'vNRCV_Mx1' in xm.dataFrames
False
>>> mx=xm.MxSync()
>>> 'vNRCV_Mx1' in xm.dataFrames
True
>>> vROHR.shape
(16, 76)
>>> print(xm._getvXXXXAsOneString(vXXXX='vROHR',dropColList=['NAME_i_2L','NAME_k_2L
→']))
    BESCHREIBUNG IDREFERENZ BAUJAHR HAL IPLANUNG KENNUNG      L LZU  RAU ZAUS ZEIN
→ZUML JLAMBS LAMBDA0 ASOLL INDSCHALL KVR AUSFALLZEIT      DA  DI  DN      KT  PN
→REHABILITATION REPARATUR      S WSTEIG WTIEFE LTGR_NAME      LTGR_BESCHREIBUNG
→SICHTBARKEIT VERLEGEART DTRO_NAME      DTRO_BESCHREIBUNG      E
→fkSTRASSE fksRAT      pk      tk IRTRENN LECKSTART
→LECKEND LECKMENGE LECKORT LECKSTATUS QSVB ZVLIMPTNZ KANTENZV
→      CONT_CONT_ID CONT_LFDNR NAME_i KVR_i TM_i  XKOR_i  YKOR_i ZKOR_i
→NAME_k KVR_k TM_k  XKOR_k  YKOR_k ZKOR_k pXCor_i pYCor_i pXCor_k pYCor_k
→      pXCors      pYCors      pWAYPXCors
→      pWAYPYCors
→      WAYP  mx2NofPts  mx2Idx
0      None      -1      None  0      1      0  88.02  0  0.1  0  0
→0      1  0.025  1000      0  2      NaN  114.3  107.1  100  0.325  NaN
→      NaN      NaN  3.6  NaN  NaN  KUMANRO Beschreibung Leitungsgruppe
→      1  999999  KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT  2.1E+11
→      -1      -1  4613782368750024999  4613782368750024999      0      0      0
→      0      0      0      0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu      1001      -1 R-K004      2  60  2541539  5706361  20 R-
→K005      2  60  2541627  5706363      20  799.0  152.0  887.0  154.0
→[799.0, 887.0] [152.0, 154.0] [807.89999999999069, 895.95000000001863] [140.
→099999999962747, 142.049999999981374]
→[(2541547.9, 5706349.1), (2541635.95, 5706351.05)]      2      0
1      None      -1      None  0      1      0  405.96  0  0.1  0  0
→0      1  0.025  1000      0  1      NaN  114.3  107.1  100  0.325  NaN
→      NaN      NaN  3.6  NaN  NaN  KUMANRO Beschreibung Leitungsgruppe
→      1  999999  KUMANRO Kunststoffmantelrohr DN20-800 PANISOVIT  2.1E+11
→      -1      -1  4614949065966596185  4614949065966596185      0      0      0
→      0      0      0      0      NaN      NaN Nahwärmenetz mit 1000 kW
→Anschlussleistu      1001      -1 V-K002      1  90  2541059  5706265  20 V-
→K003      1  90  2541457  5706345      20  319.0  56.0  717.0  136.0
→[319.0, 717.0] [56.0, 136.0] [319.0, 716.95000000001863]
→[56.04999999999813735, 136.0]
→[(2541059.0, 5706265.05), (2541456.95, 5706345.0)]      2      1

```

(continues on next page)

(continued from previous page)

2	None	-1	None	0	1	0	83.55	0	0.1	0	0	
→0	1	0.025	1000	0	2	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	4637102239750163477	4637102239750163477	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001		-1	R-K003	2	60	2541457	5706345	20	R-		
→K004	2	60	2541539	5706361	20	717.0	136.0	799.0	152.0			
→[717.0, 799.0]	[136.0, 152.0]											
→04999999981374,	140.09999999962747]											
→[(2541465.85,	5706333.05), (2541547.9,	5706349.1)]				2	2					
3	None	-1	None	0	1	0	88.02	0	0.1	0	0	
→0	1	0.025	1000	0	1	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	4713733238627697042	4713733238627697042	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001		-1	V-K004	1	90	2541539	5706361	20	V-		
→K005	1	90	2541627	5706363	20	799.0	152.0	887.0	154.0			
→[799.0, 887.0]	[152.0, 154.0]											
→												
→[(2541539.0,	5706361.0), (2541627.05,	5706363.0)]				2	3					
4	None	-1	None	0	1	0	195.53	0	0.1	0	0	
→0	1	0.025	1000	0	1	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	4789218195240364437	4789218195240364437	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001		-1	V-K001	1	90	2540867	5706228	20	V-		
→K002	1	90	2541059	5706265	20	127.0	19.0	319.0	56.0			
→[127.0, 319.0]	[19.0, 56.0]											
→												
→[(2540867.0,	5706228.0), (2541059.0,	5706265.05)]				2	5					
5	None	-1	None	0	1	0	109.77	0	0.1	0	0	
→0	1	0.025	1000	0	2	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	4945727430885351042	4945727430885351042	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001		-1	R-K006	2	60	2541790	5706338	20	R-		
→K007	2	60	2541899	5706325	20	1050.0	129.0	1159.0	116.0	[1050.		
→0, 1159.0]	[129.0, 116.0]											
→[117.0, 104.09999999962747]												
→[(2541798.85,	5706326.0), (2541907.9,	5706313.1)]				2	7					
6	None	-1	None	0	1	0	76.4	0	0.1	0	0	
→0	1	0.025	1000	0	1	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	4984202422877610920	4984202422877610920	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001		-1	V-K000	1	90	2540793	5706209	20	V-		
→K001	1	90	2540867	5706228	20	53.0	0.0	127.0	19.0			
→[53.0, 127.0]	[0.0, 19.0]											
→[-0.0499999999813735485,	19.0]											
→[(2540793.05,	5706208.95), (2540867.0,	5706228.0)]				2	8					
7	None	-1	None	0	1	0	83.55	0	0.1	0	0	
→0	1	0.025	1000	0	1	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	5037777106796980248	5037777106796980248	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001		-1	V-K003	1	90	2541457	5706345	20	V-		
→K004	1	90	2541539	5706361	20	717.0	136.0	799.0	152.0			
→[717.0, 799.0]	[136.0, 152.0]											
→[716.9500000001863,	799.0]											

(continues on next page)

(continued from previous page)

8	None	-1	None	0	1	0	164.91	0	0.1	0	0	
→0	1	0.025	1000	0	1	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	5123819811204259837	5123819811204259837	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001	-1	V-K005	1	90	2541627	5706363	20	V-			
→K006	1	90	2541790	5706338	20	887.0	154.0	1050.0	129.0	[887.		
→0, 1050.0]	[154.0, 129.0]	[887.04999999998137,	1049.9500000001863]									
→[154.0, 128.95000000018626]												
→[(2541627.05, 5706363.0), (2541789.95, 5706337.95)]	2	10										
9	None	-1	None	0	1	0	195.53	0	0.1	0	0	
→0	1	0.025	1000	0	2	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	5266224553324203132	5266224553324203132	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001	-1	R-K001	2	60	2540867	5706228	20	R-			
→K002	2	60	2541059	5706265	20	127.0	19.0	319.0	56.0			
→[127.0, 319.0]	[19.0, 56.0]	[135.899999999990687,	327.899999999990687]	[7.								
→04999999998137355, 44.099999999962747]												
→[(2540875.9, 5706216.05), (2541067.9, 5706253.1)]	2	11										
10	None	-1	None	0	1	0	405.96	0	0.1	0	0	
→0	1	0.025	1000	0	2	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	5379365049009065623	5379365049009065623	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001	-1	R-K002	2	60	2541059	5706265	20	R-			
→K003	2	60	2541457	5706345	20	319.0	56.0	717.0	136.0			
→[319.0, 717.0]	[56.0, 136.0]	[327.899999999990687,	725.8500000000931]	[44.								
→099999999962747, 124.049999999981374]												
→[(2541067.9, 5706253.1), (2541465.85, 5706333.05)]	2	12										
11	None	-1	None	0	1	0	164.91	0	0.1	0	0	
→0	1	0.025	1000	0	2	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	5611703699850694889	5611703699850694889	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001	-1	R-K005	2	60	2541627	5706363	20	R-			
→K006	2	60	2541790	5706338	20	887.0	154.0	1050.0	129.0	[887.		
→0, 1050.0]	[154.0, 129.0]	[895.9500000001863,	1058.8500000000931]									
→[142.049999999981374, 117.0]												
→[(2541635.95, 5706351.05), (2541798.85, 5706326.0)]	2	13										
12	None	-1	None	0	1	0	109.77	0	0.1	0	0	
→0	1	0.025	1000	0	1	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	5620197984230756681	5620197984230756681	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001	-1	V-K006	1	90	2541790	5706338	20	V-			
→K007	1	90	2541899	5706325	20	1050.0	129.0	1159.0	116.0	[1050.		
→0, 1159.0]	[129.0, 116.0]	[1049.9500000001863,	1159.0]	[128.								
→95000000018626, 116.049999999981374]												
→[(2541789.95, 5706337.95), (2541899.0, 5706325.05)]	2	14										
13	None	-1	None	0	1	0	76.4	0	0.1	0	0	
→0	1	0.025	1000	0	2	NaN	114.3	107.1	100	0.325	NaN	
→	NaN	NaN	3.6	NaN	NaN	KUMANRO	Beschreibung	Leitungsgruppe				
→	1	999999	KUMANRO	Kunststoffmantelrohr	DN20-800	PANISOVIT	2.1E+11					
→	-1	-1	5647213228462830353	5647213228462830353	0	0	0					
→	0	0	0	0	NaN	NaN	Nahwärmenetz mit 1000 kW					
→Anschlussleistu	1001	-1	R-K000	2	60	2540793	5706209	20	R-			
→K001	2	60	2540867	5706228	20	53.0	0.0	127.0	19.0			
→[53.0, 127.0]	[0.0, 19.0]	[61.950000000186265,	135.899999999990687]									

(continues on next page)

(continued from previous page)

```

14      None      -1      None      0      1      0      73.42      0      0.1      0      0
→0      1      0.025      1000      0      2      NaN      168.3      160.3      150      0.45      NaN
→      NaN      NaN      4      NaN      NaN      KUMANRO      Beschreibung Leitungsgruppe
→      1      999999      KUMANRO      Kunststoffmantelrohr DN20-800 PANISOVIT      2.1E+11
→      -1      -1      4769996343148550485      4769996343148550485      0      0      0
→      0      0      0      0      NaN      NaN      Nahwärmenetz mit 1000 kW
→Anschlussleistu      1001      -1      R-L      2      60      2540740      5706225      20      R-
→K000      2      60      2540793      5706209      20      0.0      16.0      53.0      0.0
→[0.0, 53.0]      [16.0, 0.0]      [0.0, 24.0, 45.0, 61.950000000186265]
→[16.0, 16.0, -12.0, -12.0]      [(2540740.0, 5706225.0), (2540764.0, 5706225.0),
→(2540785.0, 5706197.0), (2540801.95, 5706197.0)]      2      4
15      None      -1      None      0      1      0      68.6      0      0.1      0      0
→0      1      0.025      1000      0      1      NaN      168.3      160.3      150      0.45      NaN
→      NaN      NaN      4      NaN      NaN      KUMANRO      Beschreibung Leitungsgruppe
→      1      999999      KUMANRO      Kunststoffmantelrohr DN20-800 PANISOVIT      2.1E+11
→      -1      -1      4939422678063487923      4939422678063487923      0      0      0
→      0      0      0      0      NaN      NaN      Nahwärmenetz mit 1000 kW
→Anschlussleistu      1001      -1      V-L      1      90      2540740      5706240      20      V-
→K000      1      90      2540793      5706209      20      0.0      31.0      53.0      0.0
→[0.0, 53.0]      [31.0, 0.0]      [0.0, 30.0, 53.049999999813735]      [31.0,
→31.0, -0.049999999813735485]      [(2540740.0, 5706240.0),
→(2540770.0, 5706240.0), (2540793.05, 5706208.95)]      2      6
>>> # -----
>>> # MxSync()
>>> # -----
>>> xm=Xm(xmlFile=xmlFile)
>>> vROHR=xm.dataFrames['vROHR']
>>> vROHR.shape
(16, 74)
>>> 'vNRCV_Mx1' in xm.dataFrames
False
>>> (wDir,modelDir,modelName,mx1File)=xm.getWDirModelDirModelName()
>>> mx=Mx.Mx(mx1File=mx1File)
>>> xm.MxSync(mx=mx)
>>> vROHR.shape
(16, 76)
>>> 'vNRCV_Mx1' in xm.dataFrames
True
>>> # ---
>>> # vNRCV_Mx1
>>> # ---
>>> import re
>>> f=lambda s: re.match('(\S+)~(\S*)~(\S*)~(\d+)~(\S+)',s).group(1)+'~~~'+re.match(
→'(\S+)~(\S*)~(\S*)~(\d+)~(\S+)',s).group(4)+'~'+re.match('(\S+)~(\S*)~(\S*)~(\d+)~(\S+)',s).group(5)
>>> print(xm._getvXXXXAsOneString(vXXXX='vNRCV_Mx1',end=14,dropColList=['DPGR','CONT_
→LFDNR','pk_ROWS'],mapFunc={'Sir3sID':f},sortList=['Sir3sID'],fmtFunc={'Sir3sID':f},
→index=False,header=False))
      FWES~~~5638756766880678918~W      1      BHKW      1002
→FWES      5638756766880678918      W      5762106696740202356      4857294696992797631
→4857294696992797631      (90.0, 65.0)
      KNOT~~~5049461676240771430~T      1      BHKW      1002
→KNOT      5049461676240771430      T      4723443975311885965      5097127385155151127
→5097127385155151127      (90.0, 95.0)
      KNOT~~~5219230031772497417~T      1      BHKW      1002
→KNOT      5219230031772497417      T      5602301870151014230      5557806245003742769
→5557806245003742769      (90.0, 35.0)

```

(continues on next page)

(continued from previous page)

```

KNOT~~~5356267303828212700~PH 1 BHKW 1002
→KNOT 5356267303828212700 PH 5000989080893535213 4968703141722117357
→4968703141722117357 (220.0, 25.0)
KNOT~~~5397990465339071638~QM 1 Nahwärmenetz mit 1000 kW Anschlussleistu 1001
→KNOT 5397990465339071638 QM 5134531789044068877 5410059595276504750
→5410059595276504750 (91.0, -94.0)
KNOT~~~5736262931552588702~PH 1 BHKW 1002
→KNOT 5736262931552588702 PH 4754881272083464445 4681213816714574464
→4681213816714574464 (220.0, 85.0)
KNOT~~~5741235692335544560~DP 1 Nahwärmenetz mit 1000 kW Anschlussleistu 1001
→KNOT 5741235692335544560 DP 4949183695502554728 4914949875368816179
→4914949875368816179 (1234.0, 83.0)
PUMP~~~5481331875203087055~N 1 BHKW 1002
→PUMP 5481331875203087055 N 5563842594211689762 5091374651838464239
→5091374651838464239 (170.0, 45.0)
VENT~~~4678923650983295610~QM 1 BHKW 1002
→VENT 4678923650983295610 QM 5126307362398248950 5410904806390050339
→5410904806390050339 (200.0, 110.0)
WBLZ~~~4694700216019268978~WVB 1 Nahwärmenetz mit 1000 kW Anschlussleistu 1001
→WBLZ 4694700216019268978 WVB 4778244458749966216 4991097791264453745
→4991097791264453745 (354.0, -225.0)
WBLZ~~~5262603207038486299~WES 1 BHKW 1002
→WBLZ 5262603207038486299 WES 5690691957596882133 5179988968597313889
→5179988968597313889 (90.0, 155.0)
WBLZ~~~5262603207038486299~WSPEI 1 BHKW 1002
→WBLZ 5262603207038486299 WSPEI 5153847813311339683 4946584950744559030
→4946584950744559030 (90.0, 140.0)
WBLZ~~~5262603207038486299~WVB 1 BHKW 1002
→WBLZ 5262603207038486299 WVB 5214984699859365639 5281885868749421521
→5281885868749421521 (90.0, 150.0)
WBLZ~~~5262603207038486299~WVERL 1 BHKW 1002
→WBLZ 5262603207038486299 WVERL 4722863010266870887 5476262878682325254
→5476262878682325254 (90.0, 145.0)
>>> # ---
>>> # vKNOT
>>> # ---
>>> print(xm._getvXXXXAsOneString(vXXXX='vKNOT',end=1,dropColList=['LFKT_NAME','LF',
→'LF_min','LF_max','PVAR_NAME','PH','PH_min','PH_max','PZON_NAME','FSTF_NAME','STOF_
→NAME','GMIX_NAME','UTMP_NAME','2L_NAME','2L_KVR','fkHYDR','fkFQPS']))
NAME BESCHREIBUNG IDREFERENZ CONT CONT_ID
→CONT_LFDNR CONT_VKNO KTYP LFAKT QM_EIN QVAR_NAME QM QM_min QM_max KVR TE TM
→ XKOR YKOR ZKOR pk tk pXCor pYCor mx2Idx
0 R-K004 None -1 Nahwärmenetz mit 1000 kW Anschlussleistu 1001
→ -1 NaN QKON 1 0 NaN NaN NaN NaN 2 10 60
→2541539 5706361 20 4638663808856251977 4638663808856251977 799.0 152.0
→0
>>> print(xm._getvXXXXAsOneString(vXXXX='vFWVB',dropColList=['BESCHREIBUNG',
→'IDREFERENZ','W0','W','IRFV','LFK','TVL0','TRS0','LFKT','W_min','W_max','INDTR',
→'TRSK','VTYP','DPHAUS','IMBG','pk','tk','KVR_i','TM_i','XKOR_i','YKOR_i','ZKOR_i',
→'pXCor_i','pYCor_i','KVR_k','TM_k','XKOR_k','YKOR_k','ZKOR_k','pXCor_k','pYCor_k',
→'CONT','CONT_ID','CONT_LFDNR']))
WOLFK NAME_i NAME_k WBLZ mx2Idx
0 160.0 V-K002 R-K002 [BLNZ1, BLNZ1u5, BLNZ1u5u7] 0
1 200.0 V-K004 R-K004 [] 1
2 160.0 V-K005 R-K005 [BLNZ1u5, BLNZ1u5u7, BLNZ5] 2
3 160.0 V-K007 R-K007 [BLNZ1u5u7] 3
4 120.0 V-K003 R-K003 [] 4

```

(continues on next page)

(continued from previous page)

```

>>> # ---
>>> # vXXXX
>>> # ---
>>> xm.dataFrames['vVBEL_forTestOnly']=xm.dataFrames['vVBEL'].reset_
  ↳ index(inplace=False) # Multiindex to Cols
>>> print(xm._getvXXXXAsOneString(vXXXX='vVBEL_forTestOnly',index=True,dropColList=[
  ↳ 'Z_i','pk_i','CONT_i','CONT_VKNO_i','Z_k','pk_k','CONT_k','CONT_VKNO_k','IDREFERENZ
  ↳ ','tk']))

```

	OBJTYPE	LAYR	OBJID	BESCHREIBUNG	NAME_i	NAME_k
0	FWES	5638756766880678918	BHKW	- Modul - 1000 kW therm.	R3	V-1
	[Vorlauf]	0	80	0		
1	FWVB	4643800032883366034		1	V-K002	R-K002
	[Kundenanlagen]	0	NaN	0		
2	FWVB	4704603947372595298		3	V-K004	R-K004
	[Kundenanlagen]	0	NaN	1		
3	FWVB	5121101823283893406		4	V-K005	R-K005
	[Kundenanlagen]	0	NaN	2		
4	FWVB	5400405917816384862		5	V-K007	R-K007
	[Kundenanlagen]	0	NaN	3		
5	FWVB	5695730293103267172		2	V-K003	R-K003
	[Kundenanlagen]	0	NaN	4		
6	KLAP	4801110583764519435		None	R2	R3
	[Rücklauf]	0	80	0		
7	PGRP	4986517622672493603		Pumpengruppe	R-1	R3
	[]	0	NaN	0		
8	PUMP	5481331875203087055		Umwälzpumpe	R-1	R2
	[Rücklauf]	0	NaN	0		
9	ROHR	4613782368750024999		None	R-K004	R-K005
	[Rücklauf]	88.02	107.1	0		
10	ROHR	4614949065966596185		None	V-K002	V-K003
	[Vorlauf]	405.96	107.1	1		
11	ROHR	4637102239750163477		None	R-K003	R-K004
	[Rücklauf]	83.55	107.1	2		
12	ROHR	4713733238627697042		None	V-K004	V-K005
	[Vorlauf]	88.02	107.1	3		
13	ROHR	4769996343148550485		None	R-L	R-K000
	[Rücklauf]	73.42	160.3	4		
14	ROHR	4789218195240364437		None	V-K001	V-K002
	[Vorlauf]	195.53	107.1	5		
15	ROHR	4939422678063487923		None	V-L	V-K000
	[Vorlauf]	68.6	160.3	6		
16	ROHR	4945727430885351042		None	R-K006	R-K007
	[Rücklauf]	109.77	107.1	7		
17	ROHR	4984202422877610920		None	V-K000	V-K001
	[Vorlauf]	76.4	107.1	8		
18	ROHR	5037777106796980248		None	V-K003	V-K004
	[Vorlauf]	83.55	107.1	9		
19	ROHR	5123819811204259837		None	V-K005	V-K006
	[Vorlauf]	164.91	107.1	10		
20	ROHR	5266224553324203132		None	R-K001	R-K002
	[Rücklauf]	195.53	107.1	11		
21	ROHR	5379365049009065623		None	R-K002	R-K003
	[Rücklauf]	405.96	107.1	12		
22	ROHR	5611703699850694889		None	R-K005	R-K006
	[Rücklauf]	164.91	107.1	13		
23	ROHR	5620197984230756681		None	V-K006	V-K007
	[Vorlauf]	109.77	107.1	14		

(continues on next page)

(continued from previous page)

```

24 ROHR 5647213228462830353          None      R-K000  R-K001  ↵
↪ [Rücklauf]      76.4  107.1      15
25 VENT 4678923650983295610          None          V-1      V-L  ↵
↪ [Vorlauf]        0    150        0
26 VENT 4897018421024717974          None          R-L      R-1  ↵
↪ [Rücklauf]        0    150        1
27 VENT 5525310316015533093          None  PKON-Knoten      R-1  ↵
↪ [Rücklauf]        0     50        2
>>> # ---
>>> # vRART
>>> # ---
>>> print(xm._getvXXXXAsOneString(vXXXX='vRART', index=True, sortList=['INDSTD', 'NAME
↪ ]))
      NAME          BESCHREIBUNG          INDSTD_TXT  INDSTD_
↪ DWDT WSOSTD          pk NAME_KREF1 NAME_KREF2 NAME_SWVT
0  dp  Bezeichnung Regelungsart Differenzdruck Druckseite, Sollwert Tabelle      55_
↪ 1E+20      0  5552938346422332788      V-K007      R-K007      SWVT
>>> # -----
>>> # MxAdd
>>> # -----
>>> if 'vNRCV_Mx1' in xm.dataFrames:
...     del xm.dataFrames['vNRCV_Mx1'] # delete MxSync-Result to force MxSync-Call in_
↪ MxAdd
>>> oldShape=xm.dataFrames['vKNOT'].shape
>>> mx=xm.MxAdd()
>>> firstShape=xm.dataFrames['vKNOT'].shape
>>> oldShape[1]<firstShape[1]
True
>>> xm.MxAdd(mx=mx)
>>> secondShape=xm.dataFrames['vKNOT'].shape
>>> secondShape==firstShape
True
>>> xm.MxAdd(mx=mx)
>>> thirdShape=xm.dataFrames['vKNOT'].shape
>>> thirdShape==firstShape
True
>>> xm.dataFrames['vKNOT_forTestOnly']=xm.dataFrames['vKNOT'].rename(columns={'KNOT~*~
↪ ~*~PH': 'Druck'})
>>> if 'Druck' not in xm.dataFrames['vKNOT_forTestOnly']:
...     xm.dataFrames['vKNOT_forTestOnly'].rename(columns={'KNOT~*~*~*~H': 'Druck'},
↪ inplace=True)
>>> if 'Druck' not in xm.dataFrames['vKNOT_forTestOnly']:
...     xm.dataFrames['vKNOT_forTestOnly'].rename(columns={'KNOT~*~*~*~PH': 'Druck'},
↪ inplace=True) #09
>>> f = lambda x: round(x,1) if x != None else None
>>> print(xm._getvXXXXAsOneString(vXXXX='vKNOT_forTestOnly', filterColList=['mx2Idx',
↪ 'KVR', 'NAME', 'Druck'], mapFunc={'Druck':f}, index=True))
      mx2Idx KVR      NAME  Druck
0          0  2      R-K004    2.3
1          1  1      V-K002    4.0
2          2  1      V-K001    4.1
3          3  1      V-K000    4.1
4          4  2      R-K001    2.0
5          5  2      R-K003    2.3
6          6  2      R-K000    2.0
7          9  2      R-K005    2.3
8         11  2          R-L    2.0

```

(continues on next page)

(continued from previous page)

```

9      12  2      R-K002    2.1
10     13  1      V-K004    3.8
11     15  1      V-K005    3.8
12     16  2      R-K007    2.3
13     17  1      V-K006    3.8
14     18  2      R-K006    2.3
15     20  1      V-K003    3.8
16     21  1      V-L      4.1
17     22  1      V-K007    3.8
18      7  2      R2       4.3
19      8  1      V-1      4.1
20     10  2      R3       4.3
21     14  2      PKON-Knoten 2.0
22     19  2      R-1      2.0
>>> print(xm._getvXXXXAsOneString(vXXXX='vROHR',filterColList=['mx2Idx','L','KVR',
↳ 'NAME_i','NAME_k','ROHR~*~*~*~QMAV'],mapFunc={'ROHR~*~*~*~QMAV':f},sortList=['ROHR~
↳ *~*~*~QMAV','NAME_i'],index=True))
      mx2Idx      L KVR  NAME_i  NAME_k  ROHR~*~*~*~QMAV
13      15      76.4  2  R-K000  R-K001      -23.0
9       11     195.53  2  R-K001  R-K002      -23.0
14      4      73.42  2      R-L  R-K000      -23.0
10      12     405.96  2  R-K002  R-K003     -19.1
2       2      83.55  2  R-K003  R-K004     -15.4
0       0      88.02  2  R-K004  R-K005      -8.5
11      13     164.91  2  R-K005  R-K006      -3.9
5       7     109.77  2  R-K006  R-K007      -3.9
8       10     164.91  1  V-K005  V-K006       3.9
12      14     109.77  1  V-K006  V-K007       3.9
3       3      88.02  1  V-K004  V-K005       8.5
7       9      83.55  1  V-K003  V-K004      15.4
1       1     405.96  1  V-K002  V-K003      19.1
6       8      76.4  1  V-K000  V-K001      23.0
4       5     195.53  1  V-K001  V-K002      23.0
15      6      68.6  1      V-L  V-K000      23.0
>>> print(xm._getvXXXXAsOneString(vXXXX='vFWVB',filterColList=['mx2Idx','NAME_i',
↳ 'NAME_k','FWVB~*~*~*~W'],mapFunc={'FWVB~*~*~*~W':f},sortList=['FWVB~*~*~*~W','NAME_i
↳ '],index=True))
      mx2Idx  NAME_i  NAME_k  FWVB~*~*~*~W
4       4  V-K003  R-K003      120.0
0       0  V-K002  R-K002      160.0
2       2  V-K005  R-K005      160.0
3       3  V-K007  R-K007      160.0
1       1  V-K004  R-K004      200.0
>>> xm.dataFrames['vVBEL_forTestOnly2']=xm.dataFrames['vVBEL'].loc[['ROHR','FWVB'],:].
↳ reset_index(inplace=False) # Multiindex to Cols
>>> xm.dataFrames['vVBEL_forTestOnly2'].rename(columns={'KNOT~*~*~*~PH_i':'Druck_i'},
↳ inplace=True)
>>> if 'Druck_i' not in xm.dataFrames['vVBEL_forTestOnly2']:
...     xm.dataFrames['vVBEL_forTestOnly2'].rename(columns={'KNOT~*~*~*~H_i':'Druck_i
↳ '},inplace=True)
>>> if 'Druck_i' not in xm.dataFrames['vVBEL_forTestOnly2']:
...     xm.dataFrames['vVBEL_forTestOnly2'].rename(columns={'KNOT~*~*~*~PH_i':'Druck_i
↳ '},inplace=True) #09
>>> f = lambda x: round(x,1) if x != None else None
>>> print(xm._getvXXXXAsOneString(vXXXX='vVBEL_forTestOnly2',filterColList=['OBJTYPE',
↳ 'mx2Idx','L','D','NAME_i','NAME_k','Druck_i','Q'],mapFunc={'Druck_i':f,'Q':f},
↳ sortList=['OBJTYPE','NAME_i','Q'],index=False))

```

(continues on next page)

(continued from previous page)

OBJTYPE	mx2Idx	L	D	NAME_i	NAME_k	Druck_i	Q
FWVB	0	0	NaN	V-K002	R-K002	4.0	3.9
FWVB	4	0	NaN	V-K003	R-K003	3.8	3.7
FWVB	1	0	NaN	V-K004	R-K004	3.8	6.9
FWVB	2	0	NaN	V-K005	R-K005	3.8	4.6
FWVB	3	0	NaN	V-K007	R-K007	3.8	3.9
ROHR	15	76.4	107.1	R-K000	R-K001	2.0	-23.0
ROHR	11	195.53	107.1	R-K001	R-K002	2.0	-23.0
ROHR	12	405.96	107.1	R-K002	R-K003	2.1	-19.1
ROHR	2	83.55	107.1	R-K003	R-K004	2.3	-15.4
ROHR	0	88.02	107.1	R-K004	R-K005	2.3	-8.5
ROHR	13	164.91	107.1	R-K005	R-K006	2.3	-3.9
ROHR	7	109.77	107.1	R-K006	R-K007	2.3	-3.9
ROHR	4	73.42	160.3	R-L	R-K000	2.0	-23.0
ROHR	8	76.4	107.1	V-K000	V-K001	4.1	23.0
ROHR	5	195.53	107.1	V-K001	V-K002	4.1	23.0
ROHR	1	405.96	107.1	V-K002	V-K003	4.0	19.1
ROHR	9	83.55	107.1	V-K003	V-K004	3.8	15.4
ROHR	3	88.02	107.1	V-K004	V-K005	3.8	8.5
ROHR	10	164.91	107.1	V-K005	V-K006	3.8	3.9
ROHR	14	109.77	107.1	V-K006	V-K007	3.8	3.9
ROHR	6	68.6	160.3	V-L	V-K000	4.1	23.0

```

>>> # ---
>>> # Clean Up LocalHeatingNetwork Xm and Mx
>>> # ---
>>> xm.delFiles()
>>> mx.delFiles()
>>> # ---
>>> # TinyWDN
>>> # ---
>>> xmlFile=os.path.join(os.path.join(path,testDir),'TinyWDN.XML')
>>> xm=Xm(xmlFile=xmlFile)
>>> # ---
>>> # GPipe
>>> # ---
>>> xmlFile=os.path.join(os.path.join(path,testDir),'GPipe.XML')
>>> xm=Xm(xmlFile=xmlFile)

```

class `Xm.Xm(xmlFile, NoH5Read=False)`
 SIR 3S modelFile to pandas DataFrames.

Args:

- `xmlFile (str)`: SIR 3S modelFile
- **NoH5Read (bool)**:

False (default):

- An existing _and newer h5File will be read _instead of xmlFile.
- xmlFile will _not be read (it does even not have to exist)

True:

- An existing h5File will be deleted.
- xmlFile will be read.

Attributes:

- **states**
 - h5Read: True, if read from H5
- **fileNames**
 - xmlFile
 - constructed from MX during Init and Usage:
 - h5File: corresponding h5File(name) derived from xmlFile(name)
- **dataFrames**
 - dict with pandas DataFrames
 - one pandas DataFrame per SIR 3S Objecttype (i.e. KNOT, ROHR, ...)
 - keys: KNOT, ROHR, ... and vKNOT, vROHR, ...
 - **some Views as pandas DataFrames**
 - * i.e. vKNOT, vROHR, ...
 - * The Views are designed to deal with tedious groundwork.
 - * The Views are aggregated somewhat arbitrary.
 - * However: Usage of SIR 3S Modeldata is more convenient and efficient with appropriate Views.
- **pXCorZero, pYCorZero**
 - min. X aller Knoten der Netzansicht
 - min. Y dito

Raises: `XmError`

```
classmethod _xmlRoot2Dfs (root)
```

Parse root into DataFrames.

- **Return:**
 - dict with dfs with root-content

Raises: XmError

```
>>> XmlString='<data><country name="Liechtenstein"><rank>1</rank><year>2008</
year><gdppc>141100</gdppc><neighbor name="Austria" direction="E"/><neighbor
name="Switzerland" direction="W"/></country><country name="Singapore"><rank>
4</rank><year>2011</year><gdppc>59900</gdppc><neighbor name="Malaysia"
direction="N"/></country></data>'
>>> import xml.etree.ElementTree as ET
>>> root = ET.fromstring(XmlString)
>>> import Xml
>>> dfDct=Xm.Xml._xmlRoot2Dfs(root)
>>> dfDct.keys()
dict_keys(['country'])
>>> dfDct['country']
   rank  year  gdppc neighbor neighborname neighbordirection
0      1  2008  141100    None  Switzerland                W
1      4  2011   59900    None    Malaysia                N
>>> xmlAgsnLayout='<Layout><DIAGRAM><PK_DB>5556431006082193278</PK_DB><RK_DB>
5556431006082193278</RK_DB><TITLE>HP_GLD-WAA</TITLE><TYPE>3</TYPE><LINES_
PER_GAP>2</LINES_PER_GAP><AVOID_SLACK_LINE>0</AVOID_SLACK_LINE><TABLE_RA>0</IDPH_DIM_TABLE_RA><IDPH_DIM_TABLE_P
PRESSURES><WATER_PRESSURE_DIMENSION>0</WATER_PRESSURE_DIMENSION><OVERVIEW>0
</OVERVIEW><SHOW_NODE_NAMES>1</SHOW_NODE_NAMES><SHOW_KM>1</SHOW_KM><SHOW_DIAMETERS>1</SHOW_DIAMETERS><SHOW_PUMP>1</SHOW_PUMP><SHOW_KLAP>1</SHOW_KLAP>
<SHOW_REGV>1</SHOW_REGV><SHOW_VENT>0</SHOW_VENT><SHOW_BEVE>1</SHOW_BEVE>
<SHOW_BEWI>1</SHOW_BEWI><SHOW_OBEH>1</SHOW_OBEH><SHOW_STRO>1</SHOW_STRO>
<SHOW_WIND>1</SHOW_WIND><SHOW_ACT_P>1</SHOW_ACT_P><SHOW_PIPE_AXIS>1</SHOW_PIPE_AXIS>'
>>> xmlAgsnLayout=xmlAgsnLayout.replace('<SHOW_KM>1</SHOW_KM>', '<SHOW_KM>0</SHOW_KM>')
```


(continued from previous page)

5	5556431006082193278	5681287721180972670	4847471410361330583	
↪	0 0	0		┌
6	5556431006082193278	4733201276580834622	4862900383109045212	
↪	0 0	0		┌
7	5556431006082193278	5535372591188407609	4909877841151045872	
↪	0 0	0		┌
8	5556431006082193278	4688009443988778540	5592902961516490161	
↪	0 0	0		┌
9	5556431006082193278	5027825128788497349	5374847666199546461	
↪	0 0	0		┌
10	5556431006082193278	5570221034797298909	5650209335377114460	
↪	0 0	0		┌
11	5556431006082193278	5168063508501641470	5191618168634067049	
↪	0 0	0		┌

classmethod constructNewMultiindexFromCols (*df=None*, *mColNames=['OBJTYPE', 'pk']*, *mIdxNames=['OBJTYPE', 'OBJID']*)

Constructs a new Multiindex from existing cols and returns the constructed df.

Args:

- *df*: DataFrame without Multiindex
- *mColNames*: list of columns which shall be used as Multiindex; the columns must exist; the columns will be dropped
- *mIdxNames*: list of names for the indices for the Cols above

Returns:

- df with Multiindex
- empty DataFrame is returned if an Error occurs

```
>>> d = {'OBJTYPE': ['ROHR', 'VENT'], 'pk': [123, 345], 'data': ['abc', 'def']}
>>> import pandas as pd
>>> df = pd.DataFrame(data=d)
>>> from Xm import Xm
>>> df=Xm.constructNewMultiindexFromCols(df=df,mColNames=['OBJTYPE','pk'],
↪mIdxNames=['OBJTYPE','OBJID'])
>>> df['data']
OBJTYPE  OBJID
ROHR      123    abc
VENT      345    def
Name: data, dtype: object
```

classmethod constructShortestPathFromNodeList (*df=None*, *sourceCol='NAME_i'*, *targetCol='NAME_k'*, *nl=None*, *weight=None*, *query=None*, *fmask=None*, *filterNonQ0Rows=True*)

Returns a DataFrame with Edges (one per row) implementing the shortest Path over NodeList.

Args:

- **df**: DataFrame with (all) Edges (one per row)

– **adjusting/filtering/constructing (if the corresponding cols are existing) _before using df**

- * L: converted to float before usage here

- * **Q:**

- non Null Q-rows are filtered (d.h. nur Kanten mit “Wert” bei Q werden berücksichtigt bei der Pfadermittlung)
- non Q=0-rows are filtered if filterNonQ0Rows (d.h. nur durchflossene Kanten werden berücksichtigt bei der Pfadermittlung)

- **constructed:**

- QAbs

- QAbsInv (if filterNonQ0Rows)

- nl: NodeList

- weight: columnName of the weight attribute

Der kürzeste Weg zwischen zwei Knoten in einem # zusammenhängenden Graphen ist derjenige, bei dem die # Summe der Gewichte über die durchlaufenen Kanten den # kleinstmöglichen Wert annimmt.

also bei konstantem Kantengewicht die kleinste Kantenanzahl # kürzeste Weglänge: L als Gewicht # Durchflusstärkster Weg: $1 / \text{Abs}(Q)$ (Flüsse mit 0 oder Kanten ohne Flusswert müssen vorher eliminiert werden um das Kriterium berechnen zu können ... # ... birgt die Gefahr, dass es dann keinen Weg mehr gibt) # Durchflussschwächster Weg: Q

– **examples for weight:**

- * L

- * Q (QAbs,QAbsInv)

- query: query to filter vVBEL (to filter Edges) before constructing the Graph

- fmask: function to filter vVBEL (to filter Edges) before constructing the Graph

- query and fmask are used both if not None

Returns:

- df: DataFrame with Edges (one per row) implementing the shortest Path over NodeList

- empty DataFrame is returned if an error occurs

- columns

- OBJTYPE

- OBJID

- nextNODE

- compNr

- * starts with 1

- * the number of the connected component

- * 1 for all edges if all nodes in NodeList are connected

```

>>> # -q -m 0 -s constructShortestPathFromNodeList -y no -z no -w GPipes -w
↳ LocalHeatingNetwork
>>> xmlFile=ms['GPipes']
>>> from Xm import Xm
>>> xm=Xm(xmlFile=xmlFile,NoH5Read=True)
>>> xm.constructShortestPathFromNodeList(df=xm.
↳ getvVBELwithNodeAttributeAdded(),nl=['GL','GR'])
  OBJTYPE      OBJID nextNODE  compNr
0  VENT  5309992331398639768      G1      1
1  ROHR  5244313507655010738      GKS      1
2  VENT  5116489323526156845      GKD      1
3  ROHR  5114681686941855110      G3      1
4  ROHR  4979507900871287244      G4      1
5  VENT  5745097345184516675      GR      1
>>> mx=xm.MxSync()
>>> xm.MxAdd(mx=mx)
>>> xm.constructShortestPathFromNodeList(df=xm.
↳ getvVBELwithNodeAttributeAdded(),nl=['GL','GR'],weight='QAbsInv') #
↳ durchflusstärkster Weg
  OBJTYPE      OBJID nextNODE  compNr
0  VENT  5309992331398639768      G1      1
1  ROHR  5244313507655010738      GKS      1
2  VENT  5508684139418025293      GKD      1
3  ROHR  5114681686941855110      G3      1
4  ROHR  4979507900871287244      G4      1
5  VENT  5745097345184516675      GR      1
>>> ###
>>> f=lambda row: True if row.NAME_i != 'GKS' else False
>>> xm.constructShortestPathFromNodeList(df=xm.
↳ getvVBELwithNodeAttributeAdded(),nl=['GL','GKS','GKD','GR'],fmask=f)
  OBJTYPE      OBJID nextNODE  compNr
0  VENT  5309992331398639768      G1      1
1  ROHR  5244313507655010738      GKS      1
2  ROHR  5114681686941855110      G3      2
3  ROHR  4979507900871287244      G4      2
4  VENT  5745097345184516675      GR      2
>>> ###
>>> xmlFile=ms['LocalHeatingNetwork']
>>> xm=Xm(xmlFile=xmlFile,NoH5Read=True)
>>> xm.constructShortestPathFromNodeList(df=xm.
↳ getvVBELwithNodeAttributeAdded(),nl=['V-L','V-K07'])
Empty DataFrame
Columns: []
Index: []
>>> xm.constructShortestPathFromNodeList(df=xm.
↳ getvVBELwithNodeAttributeAdded(),nl=['V-L','V-K007'])
  OBJTYPE      OBJID nextNODE  compNr
0  ROHR  4939422678063487923  V-K000      1
1  ROHR  4984202422877610920  V-K001      1
2  ROHR  4789218195240364437  V-K002      1
3  ROHR  4614949065966596185  V-K003      1
4  ROHR  5037777106796980248  V-K004      1
5  ROHR  4713733238627697042  V-K005      1
6  ROHR  5123819811204259837  V-K006      1
7  ROHR  5620197984230756681  V-K007      1
>>> xm.constructShortestPathFromNodeList(df=xm.
↳ getvVBELwithNodeAttributeAdded(),nl=['V-K007','R-K007'])

```

(continues on next page)

(continued from previous page)

```

    OBJTYPE          OBJID nextNODE  compNr
0    FWVB  5400405917816384862  R-K007      1
>>> f=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2' else False
>>> xm.constructShortestPathFromNodeList(df=xm.
↳getvVBELwithNodeAttributeAdded(),nl=['V-K007','R-K007'],fmask=f)
Empty DataFrame
Columns: []
Index: []
>>> xm.constructShortestPathFromNodeList(df=xm.
↳getvVBELwithNodeAttributeAdded(),nl=['R-K007','V-K007'],query="OBJTYPE not_
↳in ['FWVB','PGRP']")
    OBJTYPE          OBJID nextNODE  compNr
0    ROHR  4945727430885351042  R-K006      1
1    ROHR  5611703699850694889  R-K005      1
2    ROHR  4613782368750024999  R-K004      1
3    ROHR  4637102239750163477  R-K003      1
4    ROHR  5379365049009065623  R-K002      1
5    ROHR  5266224553324203132  R-K001      1
6    ROHR  5647213228462830353  R-K000      1
7    ROHR  4769996343148550485    R-L      1
8    VENT  4897018421024717974    R-1      1
9    PUMP  5481331875203087055    R2      1
10   KLAP  4801110583764519435    R3      1
11   FWES  5638756766880678918    V-1      1
12   VENT  4678923650983295610    V-L      1
13   ROHR  4939422678063487923  V-K000      1
14   ROHR  4984202422877610920  V-K001      1
15   ROHR  4789218195240364437  V-K002      1
16   ROHR  4614949065966596185  V-K003      1
17   ROHR  5037777106796980248  V-K004      1
18   ROHR  4713733238627697042  V-K005      1
19   ROHR  5123819811204259837  V-K006      1
20   ROHR  5620197984230756681  V-K007      1

```

delFiles()

Deletes Files constructed by XM during Init and Usage.

_xmlRead()

Reads the SIR 3S modelFile.

- Performs fixes and basic conversions inplace the dataFrames read from modelFile: `_convertAndFix()`
- Creates some Views: `_vXXXX()`

Raises: `XmError`

```

>>> # -q -m 0 -s xmlRead -t nothing -y yes -z no -w OneLPipe
>>> xmlFile=ms['OneLPipe']
>>> import Xm
>>> xm=Xm.Xm(xmlFile)

```

FromH5 (h5File=None)

Reads all dataFrames stored in h5File into self.DataFrames.

Args:**h5File:**

- (str): the h5File(name) to be read
 - (None): self.h5File will be read
- Reads all keys.
 - Existing keys in self.dataFrames are overwritten.

Note that after .FromH5() the content of self.dataFrames may differ from the content given by an existing self.xmlFile.

Raises: XmlError

ToH5 (*h5File=None*)

Stores self.dataFrames to h5File.

Args:

h5File:

- (str): the h5File(name) to be used
 - (None): self.h5File will be used
- Stores all keys.
 - Existing keys in h5File are overwritten.

Raises: XmlError

_convertAndFix ()

Performs fixes and basic conversions inplace the dataFrames read from self.xmlFile.

- Fixes and conversions here are integrity-oriented.
- Usage-oriented conversions (i.e. pd.to_numeric and base64.b64decode) - if any - are done in the ._vXXXX-methods.
- Vorgehen in den Sichten: Anwendungs-orientierte Konvertierung von pandas Object in ein spezifisches Format nur wenn sinnvoll bzw. erforderlich

Conversions:

- , > . (converted in: SWVT_ROW, LFKT_ROW, QVAR_ROW, PVAR_ROW)

Fixes:

- **No SWVT_ROW, LFKT_ROW, QVAR_ROW, PVAR_ROW?!**
 - * SWVT, LFKT, QVAR, PVAR are constructed to
- 1st Time without Value?! (fixed in: SWVT_ROW, LFKT_ROW, QVAR_ROW, PVAR_ROW)
- Template Node(s)?!
- **in new Models constructed from SIR 3S**
 - **not all Objectattributes are written?!**
 - * KMOT/TE
 - * FWVB/LFK
 - * LTGR/BESCHREIBUNG

*** DTRO_ROWd**

- AUSFALLZEIT
- PN
- REHABILITATION
- REPARATUR
- WSTEIG, WTIEFE

*** RSLW**

- WMIN
- WMAX

– not all Objecttypes are written?!*** CONT**

- Models with no PZONs ...
- Models with no GMIXs ...
- Models with no STOfs ...
- empty WBLZ OBJs-BLOBs
- empty LAYR OBJs-BLOBs
- BESCHREIBUNG nicht in RLVG?...
- BESCHREIBUNG nicht in RADD?...

Raises: XmlError

_constructEmptyDf (cols=['DummyCol1', 'DummyCol2'])
Constructs an empty df with cols.

Args:

- cols: list of colNames

Returns: df: constructed df

Raises: XmlError

getWDirModelDirModelName ()
Returns (wDir,modelDir,modelName,mx1FileName).

Returns: (wDir,modelDir,modelName,mx1FileName)

wDir If wDir as given literally in .xmlFile is not a valid Dir or such a wDir relative to .xmlFile-Path exists the wDir relative is returned.

mx1FileName mx1FileName is assumed to be: :...WD...B...V...BZ...M... .MX1 If not existing: *.MX1 (first match is returned) If a suitable mx1File is not existing an INFO-Message is generated.

Raises: XmlError

getVersion (type='BASIS')
Returns VERSION-String i.e. Sir3S-90-10.

Args:

- type: BASIS or VARIANTE or BZ; the DATENEbenen-TYPE from which the VERSION-String is requested

Returns:

- VERSION-String i.e. Sir3S-90-10; Sir3S-90-09 is returned wenn der Versionsstring nicht ermittelt werden konnte

Raises: XmlError

```
>>> xm=xms['OneLPipe']
>>> vStr=xm.getVersion()
>>> import re
>>> m=re.search('Sir(?:P<Db3s>[DBdb3Ss]{2})-(?:P<Major>\d+)-(?:P<Minor>\d+)$',
→vStr) # i.e. Sir3S-90-10
>>> int(m.group('Major')[0])
9
```

`_getvXXXXAsOneString` (*vXXXX=None, start=0, end=-1, dropColList=None, filterColList=None, mapFunc={}, sortList=None, ascending=True, roundDct=None, fmtFunc={}, index=True, header=True*)

Returns vXXXX-Content as one String (for Doctest-Purposes).

Args:

- vXXXX: df=self.dataFrames[vXXXX]
- start
- end
- dropColList
- filterColList
- mapFunc: col:func: df[col].map(func)
- sortList
- ascending
- roundDct
- fmtFunc: col:func: passed to df.to_string(formatter=fmtFunc, ...)
- index
- header

Returns:

- df.to_string(formatter=fmtFunc, index=index, header=header)

Raises: XmlError

`_vXXXX` ()

Creates all Views.

Views created:

- **BLOB-Data**
 - vLAYR
 - vWBLZ
 - vAGSN u. vAGSN_raw

- **Timeseries**
 - vLFKT
 - vQVAR
 - vPVAR
 - vSWVT
- **Signalmodel**
 - vRUES: RUES-Nodes of R
 - vRXXX: Nodes of R but RUES-Nodes
 - vREdges: die Kanten des Knoten-Kanten-Signalmodells
 - vRSLW
 - vRART
 - vRSTN
- **Hydraulicmodel**
 - **Nodes**
 - * vVKNO: CONT-Nodes (also called Block-Nodes)
 - * vKNOT
 - * pXCorZero, pYCorZero
 - **Edges**
 - * vROHR: Pipes
 - * vFWVB: Housestations (district heating)
 - **all Edges (all; implemented Edges see vVBEL_edges)**
 - * vVBEL
- **Annotations**
 - vNRCV
 - vGTXT

Raises: XmError

_vLAYR()

One row per LAYR and OBJ.

Returns:

columns

LAYR (also called ‘Group’)

- LFDNR
- NAME

from SIR 3S OBJ BLOB collection:

- OBJTYPE: type (i.e.ROHR) of a LAYR OBJ
- OBJID: pk (or tk?!) of a LAYR OBJ

LAYR IDs

- pk, tk

ANNOTATION

- nrObjInGroup: Element Nr. in LAYR (LFDNR) - should be 1 otherwise the same OBJ occurs in the same LAYR multiple times
- nrObjtypeInGroup: Element Nr. of OBJTYPE in LAYR (LFDNR)

SORTING LFDNR,NAME,OBJTYPE,OBJID

Raises: XmError

_vWBLZ ()

One row per WBLZ and OBJ.

Returns:

columns**WBLZ**

- AKTIV
- BESCHREIBUNG
- IDIM
- NAME

from SIR 3S OBJ BLOB collection:

- OBJTYPE: type (always KNOT?!)
- OBJID: pk (or tk?!)

WBLZ IDs

- pk

SORTING NAME,pk

Raises: XmError

_vAGSN ()

One row per AGSN and OBJ.

Returns:

columns**AGSN**

- LFDNR
- NAME
- AKTIV
- from SIR 3S OBJ BLOB collection:
 - OBJTYPE: type (i.e.ROHR)
 - OBJID: pk (or tk?!)

AGSN IDs

- pk, tk

Sequence

- **Model**

- therefore nrObjIdInAgsn (see ANNOTATION below) should be the realworld sequence

ANNOTATION

- nrObjIdInAgsn: lfd.Nr. (in Schnittrihenfolge) Obj. (der Kante) in AGSN (AGSN is defined by LFDNR not by NAME)
- **nrObjIdTypeInAgsn: should be 1 determined by raw data**
 - nrObjIdTypeInAgsn>1 - if any - are not part of the view
 - the 1st occurrence is in the view
- **Layer** 0=undef bei Netztyp 21: 1=VL, 2=RL, 0=undef wenn keine BN-Trennzeile gefunden wird, wird VL angenommen und gesetzt die BN-Trennzeile wird dem VL (1) zugerechnet
- **nextNODE: node which is connected by the edge**
 - the cut-direction is defined (per cut and comp) by edge-sequence
 - the cut node-sequence ist the (longest shortest) path between the nodes of the 1st and last edge
 - in case of 1 edge cut-direction is edge-definition and cut node-sequence is edge-definition
 - the nextNODEs are the node-sequence omitting the start-node ...
 - ... nextNODE of an edge is the node connected by this edge in cut-direction; so nextNODE might be the i-node (the source-node) of the edge
 - if edge-direction is cut-direction nextNODE is the k-node (the sink-node) of the edge
- **compNr**
 - all 1 if all edges in the cut are connected
 - otherwise the compNr (starting with 1) the edge belongs to
 - the comp-Sequence is defined by the edge-sequence
 - the nodes of the 1st and last edge in cut-definition of the comp are defining the node-Sequence of the (longest shortest) path in the comp
- **parallel Edges**
 - are omitted in the cut-Result; the 1st edge in cut-definition is in the edge
- **Abzweige**
 - are omitted in the cut-Result
 - the nodes of the 1st and last edge in cut-definition are defining the node-Sequence of the (longest shortest) path (comp-wise)
 - only edges implementing this path are in the cut-Result

Raises: XmError

```

>>> xmlFile=ms['GPipes']
>>> from Xm import Xm
>>> xm=Xm(xmlFile=xmlFile,NoH5Read=True)
>>> vAGSN=xm.dataFrames['vAGSN']
>>> schnitt=vAGSN[vAGSN['NAME']=='LR']
>>> xm.dataFrames['schnitt']=schnitt.reset_index()
>>> print(xm._getvXXXXAsOneString(vXXXX='schnitt',index=True))
  index LFDNR NAME AKTIV OBJTYPE          OBJID          pk
  →
  → tk nrObjIdInAgsn nrObjIdTypeInAgsn Layer nextNODE
  → compNr
0      7      14 LR      101      VENT  5309992331398639768  5625063016896368599
  → 5625063016896368599          1          1          0          G1
  → 1
1      8      14 LR      101      ROHR  5244313507655010738  5625063016896368599
  → 5625063016896368599          2          1          0          GKS
  → 1
2      9      14 LR      101      VENT  5508684139418025293  5625063016896368599
  → 5625063016896368599          3          1          0          GKD
  → 1
3     10      14 LR      101      ROHR  5114681686941855110  5625063016896368599
  → 5625063016896368599          4          1          0          G3
  → 1
4     11      14 LR      101      ROHR  4979507900871287244  5625063016896368599
  → 5625063016896368599          5          1          0          G4
  → 1
5     12      14 LR      101      VENT  5745097345184516675  5625063016896368599
  → 5625063016896368599          6          1          0          GR
  → 1
>>> schnitt=vAGSN[vAGSN['NAME']=='LR-Lücke']
>>> xm.dataFrames['schnitt']=schnitt.reset_index()
>>> print(xm._getvXXXXAsOneString(vXXXX='schnitt',index=True))
  index LFDNR      NAME AKTIV OBJTYPE          OBJID
  → pk
  → tk nrObjIdInAgsn nrObjIdTypeInAgsn Layer nextNODE
  → compNr
0     13     16 LR-Lücke  101      VENT  5309992331398639768
  → 5630543731618051887  5630543731618051887          1
  → 0          G1          1
1     14     16 LR-Lücke  101      ROHR  5244313507655010738
  → 5630543731618051887  5630543731618051887          2
  → 0          GKS          1
2     15     16 LR-Lücke  101      ROHR  5114681686941855110
  → 5630543731618051887  5630543731618051887          3
  → 0          G3          2
3     16     16 LR-Lücke  101      ROHR  4979507900871287244
  → 5630543731618051887  5630543731618051887          4
  → 0          G4          2
4     17     16 LR-Lücke  101      VENT  5745097345184516675
  → 5630543731618051887  5630543731618051887          5
  → 0          GR          2
>>> schnitt=vAGSN[vAGSN['NAME']=='LR-Flansch']
>>> xm.dataFrames['schnitt']=schnitt.reset_index()
>>> print(xm._getvXXXXAsOneString(vXXXX='schnitt',index=True))
  index LFDNR      NAME AKTIV OBJTYPE          OBJID
  → pk
  → tk nrObjIdInAgsn nrObjIdTypeInAgsn Layer
  → nextNODE compNr
0     18     18 LR-Flansch  101      VENT  5309992331398639768
  → 5134530907542044265  5134530907542044265          1
  → 0          G1          1

```

(continues on next page)

(continued from previous page)

```

1      19      18 LR-Flansch  101      ROHR  5244313507655010738  1
→5134530907542044265  5134530907542044265  2  1
→      0      GKS      1
2      20      18 LR-Flansch  101      VENT  5508684139418025293  1
→5134530907542044265  5134530907542044265  3  1
→      0      GKD      1
3      21      18 LR-Flansch  101      ROHR  5114681686941855110  1
→5134530907542044265  5134530907542044265  4  1
→      0      G3      1
4      22      18 LR-Flansch  101      ROHR  4979507900871287244  1
→5134530907542044265  5134530907542044265  5  1
→      0      G4      1
5      24      18 LR-Flansch  101      VENT  5745097345184516675  1
→5134530907542044265  5134530907542044265  7  1
→      0      GR      1
>>> schnitt=vAGSN[vAGSN['NAME']=='LR-Parallel']
>>> xm.dataFrames['schnitt']=schnitt.reset_index()
>>> print(xm._getvXXXXAsOneString(vXXXX='schnitt',index=True))
      index LFDNR      NAME AKTIV OBJTYPE      OBJID
→      pk      tk nrObjIdInAgsn nrObjIdTypeInAgsn Layer
→nextNODE compNr
0      25      20 LR-Parallel  101      VENT  5309992331398639768  1
→4694969854935170169  4694969854935170169  1  1
→      0      G1      1
1      26      20 LR-Parallel  101      ROHR  5244313507655010738  1
→4694969854935170169  4694969854935170169  2  1
→      0      GKS      1
2      27      20 LR-Parallel  101      VENT  5116489323526156845  1
→4694969854935170169  4694969854935170169  3  1
→      0      GKD      1
3      29      20 LR-Parallel  101      ROHR  5114681686941855110  1
→4694969854935170169  4694969854935170169  5  1
→      0      G3      1
4      30      20 LR-Parallel  101      ROHR  4979507900871287244  1
→4694969854935170169  4694969854935170169  6  1
→      0      G4      1
5      31      20 LR-Parallel  101      VENT  5745097345184516675  1
→4694969854935170169  4694969854935170169  7  1
→      0      GR      1

```

getvVBELwithNodeAttributeAdded (*nodeAttribute='KVR', preserveMultiindex=False*)

Adds two nodeAttribute-Cols (not already in vVBEL) to vVBEL and returns the df.

Args:

- *nodeAttribute* (default: 'KVR'): the Node Attribute which shall be added
- *preserveMultiindex* (default: False): if True an existing Multiindex will be preserved; False: existing Index(Indices) will be col(s)

Returns:

- **df** (which might be incomplete, corrupt or empty if an error occurs; vVBEL is unchanged):

– **new Cols:**

- * *nodeAttribute_i*
- * *nodeAttribute_k*

Raises: `XmError`

```
>>> xmlFile=ms['LocalHeatingNetwork']
>>> from Xm import Xm
>>> xm=Xm(xmlFile=xmlFile,NoH5Read=True)
>>> df=xm.getvVBELwithNodeAttributeAdded()
>>> xm.dataFrames['tmp']=df
>>> print(xm._getvXXXXAsOneString(vXXXX='tmp',index=True,filterColList=[
→ 'OBJTYPE', 'OBJID', 'NAME_i', 'NAME_k', 'KVR_i', 'KVR_k']))
```

	OBJTYPE	OBJID	NAME_i	NAME_k	KVR_i	KVR_k
0	FWES	5638756766880678918	R3	V-1	2	1
1	FWVB	4643800032883366034	V-K002	R-K002	1	2
2	ROHR	5266224553324203132	R-K001	R-K002	2	2
3	ROHR	4614949065966596185	V-K002	V-K003	1	1
4	FWVB	4704603947372595298	V-K004	R-K004	1	2
5	ROHR	4637102239750163477	R-K003	R-K004	2	2
6	ROHR	4713733238627697042	V-K004	V-K005	1	1
7	FWVB	5121101823283893406	V-K005	R-K005	1	2
8	ROHR	4613782368750024999	R-K004	R-K005	2	2
9	ROHR	5123819811204259837	V-K005	V-K006	1	1
10	FWVB	5400405917816384862	V-K007	R-K007	1	2
11	ROHR	4945727430885351042	R-K006	R-K007	2	2
12	FWVB	5695730293103267172	V-K003	R-K003	1	2
13	ROHR	5379365049009065623	R-K002	R-K003	2	2
14	ROHR	5037777106796980248	V-K003	V-K004	1	1
15	KLAP	4801110583764519435	R2	R3	2	2
16	PGRP	4986517622672493603	R-1	R3	2	2
17	PUMP	5481331875203087055	R-1	R2	2	2
18	ROHR	4769996343148550485	R-L	R-K000	2	2
19	VENT	4897018421024717974	R-L	R-1	2	2
20	VENT	5525310316015533093	PKON-Knoten	R-1	2	2
21	ROHR	4789218195240364437	V-K001	V-K002	1	1
22	ROHR	4939422678063487923	V-L	V-K000	1	1
23	ROHR	4984202422877610920	V-K000	V-K001	1	1
24	ROHR	5611703699850694889	R-K005	R-K006	2	2
25	ROHR	5620197984230756681	V-K006	V-K007	1	1
26	ROHR	5647213228462830353	R-K000	R-K001	2	2
27	VENT	4678923650983295610	V-1	V-L	1	1

```
>>> len(df.columns.tolist())
20
>>> df.index.names
FrozenList([None])
>>> df=xm.getvVBELwithNodeAttributeAdded(preserveMultiindex=True)
>>> len(df.columns.tolist())
18
>>> df.index.names
FrozenList(['OBJTYPE', 'OBJID'])
```

vAGSN_Add (*nl=None, weight=None, Layer=0, AKTIV=None, NAME='NEU', query=None, fmask=None, filterNonQORows=True*)

Adds a new User-defined Cut to the Model-defined Cuts.

Arguments:

see constructShortestPathFromNodeList:

- *nl*: `NodeList` for the Cut
- *weight*: columnName of the weight attribute
- *query*: mask to filter vBEL (to filter Edges) before constructing the Graph

- fmask: function to filter vVBEL (to filter Edges) before constructing the Graph
- query and fmask are used both if not None
- Layer (to use in constructed cut)
- AKTIV (to use in constructed cut)
- NAME (to use in constructed cut): the cut will NOT be constructed if such a NAME already exists: ERROR

Returns: True if successfull False else

Raises: XmError

```
>>> xmlFile=ms['GPipes']
>>> from Xm import Xm
>>> xm=Xm(xmlFile=xmlFile,NoH5Read=True)
>>> xm.vAGSN_Add(nl=['GL','GR'])
True
>>> import pandas as pd
>>> pd.set_option('display.max_columns',None)
>>> pd.set_option('display.max_rows',None)
>>> pd.set_option('display.max_colwidth',666666)
>>> pd.set_option('display.width',66666666)
>>> xm.dataFrames['vAGSN_raw']
```

	LFDNR		NAME	AKTIV	OBJTYPE		OBJID	
→pk				tk	nrObjIdInAgsn	nrObjIdTypeInAgsn	Layer	nextNODE
→compNr								
0	2	1	Rohr	101	ROHR	5244313507655010738		
→5015814781412926392				5015814781412926392		1		1
→0			GKS	1				
1	6		V-Rohr	101	VENT	5309992331398639768		
→5396484903084432138				5396484903084432138		1		1
→0			G1	1				
2	6		V-Rohr	101	ROHR	5244313507655010738		
→5396484903084432138				5396484903084432138		2		1
→0			GKS	1				
3	8		Rohr-V	101	ROHR	4979507900871287244		
→4989935433418681990				4989935433418681990		1		1
→0			G4	1				
4	8		Rohr-V	101	VENT	5745097345184516675		
→4989935433418681990				4989935433418681990		2		1
→0			GR	1				
5	12	2	Rohre	101	ROHR	5114681686941855110		
→5748019382126004712				5748019382126004712		1		1
→0			G3	1				
6	12	2	Rohre	101	ROHR	4979507900871287244		
→5748019382126004712				5748019382126004712		2		1
→0			G4	1				
7	14		LR	101	VENT	5309992331398639768		
→5625063016896368599				5625063016896368599		1		1
→0			G1	1				
8	14		LR	101	ROHR	5244313507655010738		
→5625063016896368599				5625063016896368599		2		1
→0			GKS	1				
9	14		LR	101	VENT	5508684139418025293		
→5625063016896368599				5625063016896368599		3		1
→0			GKD	1				

(continues on next page)

(continued from previous page)

10	14	LR	101	ROHR	5114681686941855110	1	
→	5625063016896368599		5625063016896368599		4		1
→	0	G3	1				
11	14	LR	101	ROHR	4979507900871287244	1	
→	5625063016896368599		5625063016896368599		5		1
→	0	G4	1				
12	14	LR	101	VENT	5745097345184516675	1	
→	5625063016896368599		5625063016896368599		6		1
→	0	GR	1				
13	16	LR-Lücke	101	VENT	5309992331398639768	1	
→	5630543731618051887		5630543731618051887		1		1
→	0	G1	1				
14	16	LR-Lücke	101	ROHR	5244313507655010738	1	
→	5630543731618051887		5630543731618051887		2		1
→	0	GKS	1				
15	16	LR-Lücke	101	ROHR	5114681686941855110	1	
→	5630543731618051887		5630543731618051887		3		1
→	0	G3	2				
16	16	LR-Lücke	101	ROHR	4979507900871287244	1	
→	5630543731618051887		5630543731618051887		4		1
→	0	G4	2				
17	16	LR-Lücke	101	VENT	5745097345184516675	1	
→	5630543731618051887		5630543731618051887		5		1
→	0	GR	2				
18	18	LR-Flansch	101	VENT	5309992331398639768	1	
→	5134530907542044265		5134530907542044265		1		1
→	0	G1	1				
19	18	LR-Flansch	101	ROHR	5244313507655010738	1	
→	5134530907542044265		5134530907542044265		2		1
→	0	GKS	1				
20	18	LR-Flansch	101	VENT	5508684139418025293	1	
→	5134530907542044265		5134530907542044265		3		1
→	0	GKD	1				
21	18	LR-Flansch	101	ROHR	5114681686941855110	1	
→	5134530907542044265		5134530907542044265		4		1
→	0	G3	1				
22	18	LR-Flansch	101	ROHR	4979507900871287244	1	
→	5134530907542044265		5134530907542044265		5		1
→	0	G4	1				
23	18	LR-Flansch	101	VENT	5745097345184516675	1	
→	5134530907542044265		5134530907542044265		7		1
→	0	GR	1				
24	20	LR-Parallel	101	VENT	5309992331398639768	1	
→	4694969854935170169		4694969854935170169		1		1
→	0	G1	1				
25	20	LR-Parallel	101	ROHR	5244313507655010738	1	
→	4694969854935170169		4694969854935170169		2		1
→	0	GKS	1				
26	20	LR-Parallel	101	VENT	5116489323526156845	1	
→	4694969854935170169		4694969854935170169		3		1
→	0	GKD	1				
27	20	LR-Parallel	101	ROHR	5114681686941855110	1	
→	4694969854935170169		4694969854935170169		5		1
→	0	G3	1				
28	20	LR-Parallel	101	ROHR	4979507900871287244	1	
→	4694969854935170169		4694969854935170169		6		1
→	0	G4	1				

(continues on next page)

(continued from previous page)

```

29      20 LR-Parallel      101      VENT  5745097345184516675
→4694969854935170169  4694969854935170169      7      1
→      0      GR      1
30      21      NEU None      VENT  5309992331398639768
→PT3S      PT3S      1      1      0
→G1      1
31      21      NEU None      ROHR  5244313507655010738
→PT3S      PT3S      2      1      0
→GKS      1
32      21      NEU None      VENT  5116489323526156845
→PT3S      PT3S      3      1      0
→GKD      1
33      21      NEU None      ROHR  5114681686941855110
→PT3S      PT3S      4      1      0
→G3      1
34      21      NEU None      ROHR  4979507900871287244
→PT3S      PT3S      5      1      0
→G4      1
35      21      NEU None      VENT  5745097345184516675
→PT3S      PT3S      6      1      0
→GR      1
>>> # Test if the same NAME is _not_ constructed twice ...
>>> xm.vAGSN_Add(nl=['GL','GR'])
False
>>> xm.dataFrames['vAGSN_raw'].shape
(36, 12)
>>> # test if re-use works without errors ...
>>> mx=xm.MxSync()
>>> xm.MxAdd(mx=mx)
>>> # Test weight-Option ...
>>> xm.vAGSN_Add(nl=['GL','GR'],weight='QAbsInv',NAME='GL-GR w') #
→durchflusstaerksten Weg erzwingen
True
>>> df=xm.dataFrames['vAGSN_raw']
>>> df.query("LFDNR in [21,22] and nextNODE=='GKD'")
  LFDNR      NAME AKTIV OBJTYPE      OBJID      pk      tk
→nrObjIdInAgsn nrObjIdTypeInAgsn Layer nextNODE compNr
32      21      NEU None      VENT  5116489323526156845 PT3S PT3S
→ 3      1      0      GKD      1
38      22 GL-GR w None      VENT  5508684139418025293 PT3S PT3S
→ 3      1      0      GKD      1

```

_vRART()

One row per RART.

Returns:**columns:****RART**

- NAME
- BESCHREIBUNG
- INDSTD_TXT
- INDSTD (numeric)
- DWDT

RART_BZ

- WSOSTD

ID

- pk

References

- NAME_KREF1
- NAME_KREF2
- NAME_SWVT
- [NAME_RCPL] - only if RCPLs exist

sequence: Model

Raises: XmlError

__OBJJS (*dfName*, *OBJJSDecodedColName*='OBJJSDec')
Decode a column OBJJS (a BLOB containing a SIR 3S OBJ collection).

Args: *dfName*: Name of a dataFrame with column OBJJS

columns used (in self.dataFrames[*dfName*]):

- OBJJS (BLOB): i.e.: KNOT~4668229590574507160 ...
- pk: ID (of the row)
- None is returned if these columns are missing
- in this case no changes concerning column OBJJSDecodedColName in self.dataFrames[*dfName*]

OBJJSDecodedColName: colName of the decoded OBJJS; default: OBJJSDec (i.e. the BLOB is not overwritten)

Returns:

column OBJJSDecodedColName in self.dataFrames[*dfName*] set to OBJJS decoded decoded to 'XXXX~' if OBJJS was None

dfOBJJS: dataFrame with one row per OBJ in OBJJS:

columns added (compared to self.dataFrames[*dfName*]):

- OBJTYPE
- OBJID
- OBJJSDecodedColName (if not set to 'OBJJS')

rows missing (compared to self.dataFrames[*dfName*]):

- rows with OBJJS None

Raises: XmlError

```
>>> # -q -m 0 -t both -s __OBJJS -y yes -z no -w LocalHeatingNetwork
>>> xm=xms['LocalHeatingNetwork']
>>> df=xm.__OBJJS('AGSN')
>>> df['OBJJSDecStrShort']=df['OBJJSDec'].str[1:24]
>>> df[['pk', 'NAME', 'OBJJSDecStrShort', 'OBJTYPE', 'OBJID']].iloc[:3]
```

→OBJJSDecStrShort OBJTYPE

OBJID

(continues on next page)

(continued from previous page)

```

0  5252525269080005909  Netzdruckdiagramm VL/RL: BHKW - Netzende  ROHR~
→493942267806348792  ROHR  4939422678063487923
1  5252525269080005909  Netzdruckdiagramm VL/RL: BHKW - Netzende  ROHR~
→493942267806348792  ROHR  4984202422877610920
2  5252525269080005909  Netzdruckdiagramm VL/RL: BHKW - Netzende  ROHR~
→493942267806348792  ROHR  4789218195240364437

```

`_LAYOUT_XML(dfName)`

Decode a column LAYOUT_XML (a BLOB containing XML) in dfName.

Args: dfName: Name of a dataFrame with column LAYOUT_XML

columns used (in self.dataFrames[dfName]):

- LAYOUT_XML (BLOB)
- None is returned if these columns are missing

Returns: dctDfsLAYOUT: a dct with dfs with LAYOUT-content

Raises: XmlError

```

>>> # -v -m 0 -t both -s _LAYOUT_XML -y yes -z no -w LocalHeatingNetwork
>>> xm=xms['LocalHeatingNetwork']
>>> dctDfsLAYOUT=xm._LAYOUT_XML('AGSN')
>>> sorted(dctDfsLAYOUT.keys())
['DIAGRAM', 'PROFILE_LINE', 'PROFILE_LINE_COLORS', 'X_AXIS', 'Y_AXIS']
>>> dctDfsLAYOUT['PROFILE_LINE']
      FK_DIAGRAM LINE_TYPE DRUCKNIV_P LINE_COLOR LINE_COLOR_RL LINE_
→STYLE LINE_STYLE_RL
0  5252525269080005909          1          1          0          0
→ 1          5
1  5252525269080005909          8          1          255          16711680
→ 5          5
2  5252525269080005909          3          1          16711935          128
→ 5          5
>>> dctDfsLAYOUT=xm._LAYOUT_XML('SPLZ')
>>> sorted(dctDfsLAYOUT.keys())
['DIAGRAM', 'LINE', 'Y_AXIS']
>>> dctDfsLAYOUT['LINE']
      FK_DIAGRAM LINE_TYPE LABEL          DATAPOINT LINE_COLOR CONST_
→NIVEAU_VALUE LINE_STYLE DYNAMIC LINE_WIDTH FACTOR ADDEND
0  4715028732328060917          1  None  5458207635769388996  11829830
→          0          5          0          0,4          1          0

```

`_vLFKT()`

One row per Loadfactor Timeseries.

Returns:

columns

LFKT

- NAME
- BESCHREIBUNG
- INTPOL
- ZEITOPTION

SERIES

- LF: 1st Value
- LF_min
- LF_max

LFKT ID

- pk

Raises: XmError

_vNRCV ()

One row per NRCV (NumeRiCal Value).

Returns:

columns

ANNOTATIONS

- cRefLfIdNr

CONT

- CONT
- CONT_ID
- CONT_LFDNR

DP**Datapointgroup**

- DPGR

Datapoint

- OBJTYPE
- fkOBJTYPE
- ATTRTYPE

Datapoint IDs

- pk_ROWS (pk from DPKT ab 90-10)
- tk_ROWS (tk from DPKT ab 90-10)

NRCV IDs

- pk
- tk

PLot Coordinates

- pXYLB: (X,Y): Left,Bottom

Raises: XmError

_vGTXt ()

One row per GTXt (Graphic TeXt).

Returns:

columns

GTXT

- GRAFTEXT (the text)

CONT

- CONT
- CONT_ID
- CONT_LFDNR

GTXT IDs

- pk
- tk

PLot Coordinates

- pXYLB: (X,Y): Left,Bottom

Raises: XmError

_vSWVT ()

One row per Timeseries.

Returns:

columns**SWVT**

- NAME
- BESCHREIBUNG
- INTPOL
- ZEITOPTION

SERIES

- W: 1st Value
- W_min
- W_max

SWVT ID

- pk

Raises: XmError

_vRUES ()

One row per RUES (per Übergangssymbol).

1 Zeile für jede Definition(jeden Eingang). 1 Zeile für jede Referenz einer Definition (jeden Ausgang). Der Eingang muss nicht mit einem Signal versorgt sein. Der Ausgang muss nicht per Signal verwendet sein.

Returns:

columns**RUES**

- IDUE: Name (eindeutige ID) der Übergangsstelle; bei IOTYP=3: IDUE kann offenbar undefiniert sein oder einen anderen nicht nachvollziehbaren Wert tragen; eingebbar in der GUI ist IDUE nur bei IOTYP=1
- IOTYP: 0=undefiniert|1=Eingang|3=Ausgang
- rkRUES: bei IOTYP=3: Verweis auf referenzierte Übergangsstelle; -1 sonst
- IDUE_rkRUES: bei IOTYP=3: Name (eindeutige ID) der referenzierten Übergangsstelle
- IOTYP_rkRUES: bei IOTYP=3: Typ der referenzierten Übergangsstelle
- Kn: Knotenname der RUES im Sinne eines Knoten-Kanten-Modells; IDUE bei IOTYP=1 und IDUE_rkRUES bei IOTYP=3

CONT

- CONT
- ID
- CONT_rkRUES: CONT der referenzierten Übergangsstelle
- ID_rkRUES: CONT ID der referenzierten Übergangsstelle

RUES IDs

- pk
- tk

Raises: `XmError`

```
>>> import pandas as pd
>>> # ---
>>> xm=xms['DHNetwork']
>>> # ---
>>> vRUES=xm.dataFrames['vRUES']
>>> pd.set_option('display.width', 333)
>>> pd.set_option('display.max_columns', None)
>>> pd.set_option('display.max_rows', None)
>>> vRUES.sort_values(by=['Kn', 'IOTYP']) [[ # pro Kn oten steht zuerst die_
↳Definition, dann die Referenz(en)
...     'Kn'
...     , 'IOTYP'
...     , 'IDUE' # zur Kontrolle
...     , 'CONT'
...     ]].sort_index()
      Kn IOTYP      IDUE      CONT
0  Leck_1_Ein    1  Leck_1_Ein  AGFW Symposium DH
1  Leck_2_Ein    1  Leck_2_Ein  AGFW Symposium DH
2  Leck_3_Ein    1  Leck_3_Ein  AGFW Symposium DH
3          wNA    1          wNA  AGFW Symposium DH
4          wNB    1          wNB  AGFW Symposium DH
5          wNC    1          wNC  AGFW Symposium DH
6  vorOrtNA    1  vorOrtNA  AGFW Symposium DH
7  wDH_RD_A    1  wDH_RD_A  AGFW Symposium DH
8  wDH_MD_A    1  wDH_MD_A  AGFW Symposium DH
9  wDH_BA_A_RD    1  wDH_BA_A_RD  AGFW Symposium DH
10 wDH_BA_A_RD    3          NaN  AGFW Symposium DH
11 wDH_BA_A_MD    1  wDH_BA_A_MD  AGFW Symposium DH
12          dpA    1          dpA  AGFW Symposium DH
13          qB    1          qB  AGFW Symposium DH
```

(continues on next page)

(continued from previous page)

14	qC	1	qC	AGFW Symposium DH
15	vorOrtNA	3	NaN	AGFW Symposium DH
16	wNAEin	1	wNAEin	AGFW Symposium DH
17	vorOrtNB	1	vorOrtNB	AGFW Symposium DH
18	vorOrtNB	3	NaN	AGFW Symposium DH
19	wNBEin	1	wNBEin	AGFW Symposium DH
20	vorOrtNC	1	vorOrtNC	AGFW Symposium DH
21	vorOrtNC	3	NaN	AGFW Symposium DH
22	wNCEin	1	wNCEin	AGFW Symposium DH
23	wLast	1	wLast	AGFW Symposium DH
24	wTRST	1	wTRST	AGFW Symposium DH
25	Leck_Menge	1	Leck_Menge	AGFW Symposium DH
26	Leck_VL	1	Leck_VL	AGFW Symposium DH
27	Leck_RL	1	Leck_RL	AGFW Symposium DH
28	0	1	0	Diverse Steuerungen
29	1	1	1	Diverse Steuerungen
30	ADum	1	ADum	Diverse Steuerungen
31	ADum	3	NaN	Diverse Steuerungen
32	ADum	3	NaN	Diverse Steuerungen
33	ADum	3	NaN	Diverse Steuerungen
34	Leck_Menge	3	NaN	Diverse Steuerungen
35	Leck_Menge	3	NaN	Diverse Steuerungen
36	Leck_Menge	3	NaN	Diverse Steuerungen
37	Leck_Menge	3	NaN	Diverse Steuerungen
38	Leck_Menge	3	NaN	Diverse Steuerungen
39	Leck_Menge	3	NaN	Diverse Steuerungen
40	1	3	NaN	Diverse Steuerungen
41	1	3	NaN	Diverse Steuerungen
42	1	3	NaN	Diverse Steuerungen
43	1	3	NaN	Diverse Steuerungen
44	1	3	NaN	Diverse Steuerungen
45	1	3	NaN	Diverse Steuerungen
46	ADum	3	ADum	Diverse Steuerungen
47	0	3	NTRx1xEin	Diverse Steuerungen
48	0	3	NTRx1xEin	Diverse Steuerungen
49	ADum	3	ADum	Diverse Steuerungen
50	0	3	NTRx1xAus	Diverse Steuerungen
51	ADum	3	ADum	Diverse Steuerungen
52	0	3	NTRx1xAus	Diverse Steuerungen
53	ADum	3	ADum	Diverse Steuerungen
54	0	3	NTRx3xAus	Diverse Steuerungen
55	ADum	3	ADum	Diverse Steuerungen
56	0	3	NTRx3xAus	Diverse Steuerungen
57	ADum	3	ADum	Diverse Steuerungen
58	0	3	NTRx3xEin	Diverse Steuerungen
59	ADum	3	ADum	Diverse Steuerungen
60	0	3	NTRx3xEin	Diverse Steuerungen
61	ADum	3	ADum	Diverse Steuerungen
62	0	3	NTRx2xAus	Diverse Steuerungen
63	ADum	3	ADum	Diverse Steuerungen
64	0	3	NTRx2xAus	Diverse Steuerungen
65	ADum	3	ADum	Diverse Steuerungen
66	0	3	NTRx2xEin	Diverse Steuerungen
67	ADum	3	ADum	Diverse Steuerungen
68	0	3	NTRx2xEin	Diverse Steuerungen
69	ADum	3	ADum	Diverse Steuerungen
70	yDH_dp2_A	3	NaN	Diverse Steuerungen

(continues on next page)

(continued from previous page)

71	wDH_MD_A	3	NaN	Diverse Steuerungen
72	wDH_MD_A_ERO	1	wDH_MD_A_ERO	Diverse Steuerungen
73	wDH_RD_A	3	NaN	Diverse Steuerungen
74	wDH_RD_A_ERO	1	wDH_RD_A_ERO	Diverse Steuerungen
75	wDH_RD_A_ERO	3	NaN	Diverse Steuerungen
76	wDH_BA_A_RD	3	NaN	Diverse Steuerungen
77	wDH_MD_A_ERO	3	NaN	Diverse Steuerungen
78	wDH_BA_A_MD	3	NaN	Diverse Steuerungen
79	wLast	3	NaN	Diverse Steuerungen
80	1	3	NaN	Diverse Steuerungen
81	wTRST	3	NaN	Diverse Steuerungen
82	1	3	NaN	Diverse Steuerungen
83	100	1	100	Diverse Steuerungen
84	Leck_1_Ein	3	NaN	Diverse Steuerungen
85	Leck_VL	3	NaN	Diverse Steuerungen
86	ADum	3	NaN	Diverse Steuerungen
87	ADum	3	NaN	Diverse Steuerungen
88	Leck_1_Ein	3	NaN	Diverse Steuerungen
89	Leck_RL	3	NaN	Diverse Steuerungen
90	ADum	3	NaN	Diverse Steuerungen
91	ADum	3	NaN	Diverse Steuerungen
92	Leck_RL	3	NaN	Diverse Steuerungen
93	Leck_2_Ein	3	NaN	Diverse Steuerungen
94	Leck_2_Ein	3	NaN	Diverse Steuerungen
95	Leck_VL	3	NaN	Diverse Steuerungen
96	ADum	3	NaN	Diverse Steuerungen
97	ADum	3	NaN	Diverse Steuerungen
98	ADum	3	NaN	Diverse Steuerungen
99	ADum	3	NaN	Diverse Steuerungen
100	Leck_RL	3	NaN	Diverse Steuerungen
101	Leck_3_Ein	3	NaN	Diverse Steuerungen
102	Leck_3_Ein	3	NaN	Diverse Steuerungen
103	Leck_VL	3	NaN	Diverse Steuerungen
104	ADum	3	NaN	Diverse Steuerungen
105	wNAEin	3	NaN	A
106	wNA	3	NaN	A
107	wNBEin	3	NaN	A
108	ADum	3	NaN	A
109	vorOrtNC	3	NaN	A
110	ADum	3	NaN	A
111	dpA	3	NaN	A
112	vorOrtNC	3	NaN	A
113	wNBEin	3	NaN	B
114	wNB	3	NaN	B
115	wNBEin	3	NaN	B
116	ADum	3	NaN	B
117	vorOrtNB	3	NaN	B
118	ADum	3	NaN	B
119	qB	3	NaN	B
120	vorOrtNB	3	NaN	B
121	wNCEin	3	NaN	C
122	wNC	3	NaN	C
123	wNCEin	3	NaN	C
124	ADum	3	NaN	C
125	ADum	3	NaN	C
126	vorOrtNC	3	NaN	C
127	qC	3	NaN	C

(continues on next page)

(continued from previous page)

```

128     vorOrtNC      3      NaN      C
129     QDHGes       1      QDHGes    Sekundärwerte
130     yDH_dp2_A    1      yDH_dp2_A  Sekundärwerte
131     yDH_dp2_A    3      NaN      Sekundärwerte
132     yDH_pMD_A    1      yDH_pMD_A  Sekundärwerte
133     yDH_pRL_A    1      yDH_pRL_A  Sekundärwerte
134     yDH_pRL_A    3      NaN      Sekundärwerte
135     yUWM         1      yUWM      Sekundärwerte
136     wLastMW      1      wLastMW    Sekundärwerte
137     yLastMW      1      yLastMW    Sekundärwerte
138     yLastMW      3      NaN      Sekundärwerte
139     wLastMW      3      NaN      Sekundärwerte
140     dLastMW      1      dLastMW    Sekundärwerte
141     yAMW         1      yAMW      Sekundärwerte
142     yBMW         1      yBMW      Sekundärwerte
143     yCMW         1      yCMW      Sekundärwerte
144     yLastMW      3      NaN      Sekundärwerte
145     yAMW         3      NaN      Sekundärwerte
146     100          3      NaN      Sekundärwerte
147     yLastMW      3      NaN      Sekundärwerte
148     yBMW         3      NaN      Sekundärwerte
149     100          3      NaN      Sekundärwerte
150     yLastMW      3      NaN      Sekundärwerte
151     yCMW         3      NaN      Sekundärwerte
152     100          3      NaN      Sekundärwerte
153     dUWMMin     1      dUWMMin    Sekundärwerte
154     dUWMMax     1      dUWMMax    Sekundärwerte
155     yUWMLast    1      yUWMLast    Sekundärwerte
156     yUWM        3      NaN      Sekundärwerte

>>> # ---
>>> vRUESDefs=vRUES.loc[vRUES['IOTYP']=='1']
>>> # für Defs die Originaldefinition finden ...
>>> vRUESDefsCrgl=pd.merge(vRUESDefs,xm.dataFrames['CRGL'],left_on='pk',
→right_on='fkKk',suffixes=('_', '_CRGL'),how='left') # für alle sollte eine_
→Referenz gefunden werden ...
>>> vRUESDefsCrgl.sort_values(by=['Kn']) [
...     'Kn'
...     , 'CONT'
...     , 'fkKi'
...     ]

      Kn      CONT      fkKi
24      0  Diverse Steuerungen  5486870913514090048
25      1  Diverse Steuerungen  5377084992102722959
29     100  Diverse Steuerungen  5055797784689898209
26     ADum  Diverse Steuerungen  5408457159782566744
0     Leck_1_Ein  AGFW Symposium DH  5706111677806224290
1     Leck_2_Ein  AGFW Symposium DH  4704869532416514405
2     Leck_3_Ein  AGFW Symposium DH  4808434710442736644
21    Leck_Menge  AGFW Symposium DH  5390061625789905096
23     Leck_RL    AGFW Symposium DH  5644481773793849108
22     Leck_VL    AGFW Symposium DH  4880440884169110259
30     QDHGes     Sekundärwerte  5345716897595312355
37     dLastMW     Sekundärwerte  4611793887272861500
42     dUWMMax     Sekundärwerte  4672771372882677276
41     dUWMMin     Sekundärwerte  5463544828758888616
11      dpA       AGFW Symposium DH  4849866990207957614
12      qB        AGFW Symposium DH  4771725364091629759

```

(continues on next page)

(continued from previous page)

```

13      qC      AGFW Symposium DH 4978409087288292434
6      vorOrtNA AGFW Symposium DH 5194343043762135519
15     vorOrtNB AGFW Symposium DH 4705080808435797677
17     vorOrtNC AGFW Symposium DH 5620348872583735825
10    wDH_BA_A_MD AGFW Symposium DH 4873987359791313088
9     wDH_BA_A_RD AGFW Symposium DH 5322890886142492590
8     wDH_MD_A   AGFW Symposium DH 5093705160009582980
27    wDH_MD_A_ERO Diverse Steuerungen 5729434727271745948
7     wDH_RD_A   AGFW Symposium DH 4622192786925004485
28    wDH_RD_A_ERO Diverse Steuerungen 4980847179402621205
19     wLast     AGFW Symposium DH 5741660563170722352
35     wLastMW   Sekundärwerte 4833634373103605497
3     wNA        AGFW Symposium DH 4991855568438544033
14     wNAEin    AGFW Symposium DH 4742316320267545359
4     wNB        AGFW Symposium DH 4658075570394029953
16     wNBEin    AGFW Symposium DH 5013654033692161674
5     wNC        AGFW Symposium DH 5240575308071562858
18     wNCEin    AGFW Symposium DH 5670691593026035398
20     wTRST     AGFW Symposium DH 5547011912763631199
38     yAMW      Sekundärwerte 4726758453134789052
39     yBMW      Sekundärwerte 5528896084200811302
40     yCMW      Sekundärwerte 5274276049082272588
31     yDH_dp2_A Sekundärwerte 5512879293670562022
32     yDH_pMD_A Sekundärwerte 5255402486218254174
33     yDH_pRL_A Sekundärwerte 4639451967914783278
36     yLastMW   Sekundärwerte 4817923247686815456
34     yUWM      Sekundärwerte 5008805081156446169
43     yUWMLast  Sekundärwerte 5574611204646558662
>>> vRUESDefsCrglRuesDef=pd.merge(vRUESDefsCrgl,vRUES,left_on='fkKi',right_
↳on='pk',suffixes=('_', '_vRUES'),how='inner') # für die RUES-definierten_
↳RUES sollte eine Referenz gefunden werden ...
>>> vRUESDefsCrglRuesDef.sort_values(by=['Kn']) [[
...      'Kn'
...      , 'CONT'
...      , 'fkKi'
...      , 'Kn_vRUES'
...      ]]
      Kn      CONT      fkKi  Kn_vRUES
0  wDH_RD_A_ERO  Diverse Steuerungen 4980847179402621205  wDH_RD_A

```

_vRSLW (vSWVT=None)

One row per RSLW.

Args:

- vSWVT

Returns:**columns****RSLW**

- KA
- BESCHREIBUNG
- INDWBG
- WMIN

- WMAX
- INDWNO

RSLW BZ

- INDSLW
- SLWKON

CONT

- CONT
- ID
- CONT_PARENT

SWVT

- SWVT
- SWVT_Count (Anzahl der RSLW-Referenzierungen der SWVT; 0, wenn keine SWVT angegeben; INDSLW wird bei der Ermittlung nicht ausgewertet)
- BESCHREIBUNG_SWVT
- INTPOL
- ZEITOPTION

SERIES

- W: 1st Value
- W_min
- W_max

RSLW IDs

- pk
- tk

Raises: `XmError`

```
>>> import pandas as pd
>>> # ---
>>> xm=xms ['DHNetwork']
>>> # ---
>>> vRSLW=xm.dataFrames ['vRSLW']
>>> vRSLW[ [
...   'KA'
... , 'BESCHREIBUNG'
... , 'INDSLW'
... , 'CONT'
... , 'CONT_PARENT'
... , 'SWVT'
... ] ].sort_values (by= ['KA'])
```

	KA	BESCHREIBUNG	INDSLW	CONT
→CONT_PARENT		SWVT		
20	0	None	0	Diverse Steuerungen AGFW
→Symposium DH		NaN		
21	1	None	0	Diverse Steuerungen AGFW
→Symposium DH		NaN		

(continues on next page)

(continued from previous page)

23	100	None	0	Diverse Steuerungen	AGFW_
→Symposium DH	NaN				
22	ADum	Analog Dummy	0	Diverse Steuerungen	AGFW_
→Symposium DH	NaN				
0	Leck_1_Ein	Leck_1_Ein	1	AGFW Symposium DH	AGFW_
→Symposium DH	zLeck1				
1	Leck_2_Ein	Leck_2_Ein	1	AGFW Symposium DH	AGFW_
→Symposium DH	zLeck2				
2	Leck_3_Ein	Leck_3_Ein	1	AGFW Symposium DH	AGFW_
→Symposium DH	zLeck3				
11	Leck_Menge	Leck_Menge	1	AGFW Symposium DH	AGFW_
→Symposium DH	zLeckMenge				
13	Leck_RL	Leck_RL	1	AGFW Symposium DH	AGFW_
→Symposium DH	zLeck_RL				
12	Leck_VL	Leck_VL	1	AGFW Symposium DH	AGFW_
→Symposium DH	zLeck_VL				
24	cp	NaN	0	Sekundärwerte Diverse_	
→Steuerungen	NaN				
17	dpA	dpA	1	AGFW Symposium DH	AGFW_
→Symposium DH	dpA				
18	qB	qB	1	AGFW Symposium DH	AGFW_
→Symposium DH	qB				
19	qC	qC	1	AGFW Symposium DH	AGFW_
→Symposium DH	qC				
6	vorOrtNA	vorOrtNA	1	AGFW Symposium DH	AGFW_
→Symposium DH	vorOrtNA				
7	vorOrtNB	vorOrtNB	1	AGFW Symposium DH	AGFW_
→Symposium DH	vorOrtNB				
8	vorOrtNC	vorOrtNC	1	AGFW Symposium DH	AGFW_
→Symposium DH	vorOrtNC				
16	wDH_BA_A_RD	wDH_BA_A; 1=RD; 0=MD	1	AGFW Symposium DH	AGFW_
→Symposium DH	wDH_BA_A				
15	wDH_MD_A	wDH_MD_A	1	AGFW Symposium DH	AGFW_
→Symposium DH	wDH_MD_A				
14	wDH_RD_A	wDH_RD_A	1	AGFW Symposium DH	AGFW_
→Symposium DH	wDH_RD_A				
9	wLast	wLast	1	AGFW Symposium DH	AGFW_
→Symposium DH	wLast				
3	wNA	wNA	1	AGFW Symposium DH	AGFW_
→Symposium DH	wNA				
4	wNB	wNB	1	AGFW Symposium DH	AGFW_
→Symposium DH	wNB				
5	wNC	wNC	1	AGFW Symposium DH	AGFW_
→Symposium DH	wNC				
10	wTRST	wTRST	1	AGFW Symposium DH	AGFW_
→Symposium DH	wTRSP				

_vRSTN (*vSWVT=None*)

One row per RSTN.

Returns: columns: see code**Raises:** `XmError`

```

>>> # -q -m 0 -s vRSTN -w DHNetwork -y yes -z no
>>> import pandas as pd
>>> # ---

```

(continues on next page)

(continued from previous page)

```

>>> xm=xms['DHNetwork']
>>> # ---
>>> vRSTN=xm.dataFrames['vRSTN']
>>> vRSTN['RART_TYP']=vRSTN['RART_TYP'].str[:10]+'...' # zu lange_
    ↳Ausgabezeile vermeiden
>>> pd.set_option('display.width', 333)
>>> pd.set_option('display.max_columns',None)
>>> pd.set_option('display.max_rows',None)
>>> vRSTN[[
...     'CONT'
...     , 'CONT_PARENT'
...     , 'KA'
...     , 'BESCHREIBUNG'
...     , 'ITYP_OBJTYPE'
...     , 'ITYP_OBJATTR'
...     , 'Chk'
...     , 'ik_Chk'
...     , 'OBJTYPE'
...     , 'NAME_i'
...     , 'NAME_k'
...     , 'CONT_i'
...     , 'TABL_Chk'
...     , 'TABL'
...     , 'KNOT'
...     , 'RART'
...     , 'RART_TYP'
...     , 'RARTPG'
...     , 'RCPL'
...     , 'RCPL_KNOT1'
...     , 'RCPL_KNOT2'
...     , 'NAME_i_PUMP'
...     , 'NAME_k_PUMP'
... ]].sort_values(by=['ITYP_OBJTYPE','ITYP_OBJATTR','CONT','KA']).sort_
    ↳index()

```

			CONT		CONT_PARENT		KA	BESCHREIBUNG	ITYP_	
	OBJTYPE	ITYP_OBJATTR	Chk	ik_Chk	OBJTYPE	NAME_i	NAME_k			
	CONT_i	TABL_Chk	TABL	KNOT	RART		RART_TYP	RARTPG	RCPL	RCPL_
	KNOT1	RCPL_KNOT2	NAME_i_PUMP	NAME_k_PUMP						
0			A	AGFW Symposium DH	wNA_RSTN			NaN		
↳PUMP		N	1	1.0	PUMP	R-A-SS	R-A-DS			
↳A	NaN	NaN		NaN	NaN		NaN	NaN	NaN	
↳NaN		NaN		NaN	NaN					
1			A	AGFW Symposium DH	KA-0046			NaN		
↳RART		SOLL	1	NaN	NaN	NaN	NaN			
↳NaN	NaN	NaN		NaN	A_dpdS	Differenzd...		NaN	NaN	
↳NaN		NaN		NaN	NaN					
2			A	AGFW Symposium DH	KA-0044			NaN		
↳PGRP		DEAKT	1	1.0	PGRP	R-A-SS	R-A-DS-2			
↳A	NaN	NaN		NaN	NaN		NaN	NaN	NaN	
↳NaN		NaN		NaN	NaN					
3			A	AGFW Symposium DH	KA-0045			NaN		
↳PGRP		AKTIV	1	1.0	PGRP	R-A-SS	R-A-DS-2			
↳A	NaN	NaN		NaN	NaN		NaN	NaN	NaN	
↳NaN		NaN		NaN	NaN					
4			B	AGFW Symposium DH	wNB_RSTN			NaN		
↳PUMP		N	1	1.0	PUMP	R-B-SS	R-B-DS			
↳B	NaN	NaN		NaN	NaN		NaN	NaN	NaN	
↳NaN		NaN		NaN	NaN					

(continues on next page)

(continued from previous page)

5			B	AGFW Symposium DH	KA-0053		NaN		
→PGRP	DEAKT	1	1.0	PGRP	R-B-SS	R-B-DS-2			
→B	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
6			B	AGFW Symposium DH	KA-0057		NaN		
→PGRP	AKTIV	1	1.0	PGRP	R-B-SS	R-B-DS-2			
→B	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
7			B	AGFW Symposium DH	KA-0058		NaN		
→RART	SOLL	1	NaN	NaN	NaN	NaN			
→NaN	NaN NaN		NaN	B_Menge	Durchfluss...	NaN NaN			
→NaN	NaN		NaN	NaN					
8			C	AGFW Symposium DH	wNC_RSTN		NaN		
→PUMP	N	1	1.0	PUMP	R-C-SS	R-C-DS			
→C	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
9			C	AGFW Symposium DH	KA-0059		NaN		
→PGRP	DEAKT	1	1.0	PGRP	R-C-SS	R-C-DS-2			
→C	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
10			C	AGFW Symposium DH	KA-0060		NaN		
→PGRP	AKTIV	1	1.0	PGRP	R-C-SS	R-C-DS-2			
→C	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
11			C	AGFW Symposium DH	KA-0061		NaN		
→RART	SOLL	1	NaN	NaN	NaN	NaN			
→NaN	NaN NaN		NaN	C_Menge	Durchfluss...	NaN NaN			
→NaN	NaN		NaN	NaN					
12	Diverse Steuerungen			AGFW Symposium DH	KA-0004		NaN		
→LFKT	SOLL	1	NaN	NaN	NaN	NaN			
→NaN	1.0 LFKT		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
13	Diverse Steuerungen			AGFW Symposium DH	KA-0005		NaN		
→TEVT	SOLL	1	NaN	NaN	NaN	NaN			
→NaN	2.0 TRST		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
14	Diverse Steuerungen			AGFW Symposium DH	KA-0006		NaN		
→ROHR	LECKEIN	1	1.0	ROHR	V-1905	V-1906	AGFW Symposium		
→DH	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
15	Diverse Steuerungen			AGFW Symposium DH	KA-0008		NaN		
→ROHR	LECKAUS	1	1.0	ROHR	V-1905	V-1906	AGFW Symposium		
→DH	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
16	Diverse Steuerungen			AGFW Symposium DH	KA-0003		NaN		
→ROHR	LECKEIN	1	1.0	ROHR	R-1905	R-1906	AGFW Symposium		
→DH	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
17	Diverse Steuerungen			AGFW Symposium DH	KA-0007		NaN		
→ROHR	LECKAUS	1	1.0	ROHR	R-1905	R-1906	AGFW Symposium		
→DH	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
18	Diverse Steuerungen			AGFW Symposium DH	KA-0013		NaN		
→ROHR	LECKEIN	1	1.0	ROHR	V-1110	V-1111	AGFW Symposium		
→DH	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					
19	Diverse Steuerungen			AGFW Symposium DH	KA-0014		NaN		
→ROHR	LECKEIN	1	1.0	ROHR	R-1110	R-1111	AGFW Symposium		
→DH	NaN NaN		NaN	NaN		NaN NaN NaN			
→NaN	NaN		NaN	NaN					

(continues on next page)

(continued from previous page)

20	Diverse Steuerungen	AGFW Symposium DH	KA-0015	NaN						
→ROHR	LECKAUS	1	1.0	ROHR	V-1110	V-1111	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
21	Diverse Steuerungen	AGFW Symposium DH	KA-0016	NaN						
→ROHR	LECKAUS	1	1.0	ROHR	R-1110	R-1111	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
22	Diverse Steuerungen	AGFW Symposium DH	KA-0021	NaN						
→ROHR	LECKEIN	1	1.0	ROHR	V-3008	V-3007	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
23	Diverse Steuerungen	AGFW Symposium DH	KA-0022	NaN						
→ROHR	LECKEIN	1	1.0	ROHR	R-3008	R-3007	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
24	Diverse Steuerungen	AGFW Symposium DH	KA-0023	NaN						
→ROHR	LECKAUS	1	1.0	ROHR	V-3008	V-3007	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
25	Diverse Steuerungen	AGFW Symposium DH	KA-0024	NaN						
→ROHR	LECKAUS	1	1.0	ROHR	R-3008	R-3007	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
26	Diverse Steuerungen	AGFW Symposium DH	KA-0025	NaN						
→ROHR	LECKMENGE	1	1.0	ROHR	V-1905	V-1906	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
27	Diverse Steuerungen	AGFW Symposium DH	KA-0027	NaN						
→ROHR	LECKMENGE	1	1.0	ROHR	R-1905	R-1906	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
28	Diverse Steuerungen	AGFW Symposium DH	KA-0028	NaN						
→ROHR	LECKMENGE	1	1.0	ROHR	V-1110	V-1111	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
29	Diverse Steuerungen	AGFW Symposium DH	KA-0029	NaN						
→ROHR	LECKMENGE	1	1.0	ROHR	R-1110	R-1111	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
30	Diverse Steuerungen	AGFW Symposium DH	KA-0030	NaN						
→ROHR	LECKMENGE	1	1.0	ROHR	V-3008	V-3007	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
31	Diverse Steuerungen	AGFW Symposium DH	KA-0031	NaN						
→ROHR	LECKMENGE	1	1.0	ROHR	R-3008	R-3007	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
32	Diverse Steuerungen	AGFW Symposium DH	KA-0032	NTR_1_RL_Ein						
→ROHR	ZU	1	1.0	ROHR	R-1905	R-1906	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
33	Diverse Steuerungen	AGFW Symposium DH	KA-0033	NTR_1_VL_Ein						
→ROHR	ZU	1	1.0	ROHR	V-1905	V-1906	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						
34	Diverse Steuerungen	AGFW Symposium DH	KA-0034	NTR_1_VL_Ein						
→ROHR	AUF	1	1.0	ROHR	V-1905	V-1906	AGFW Symposium			
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN		
→NaN	NaN		NaN	NaN						

(continues on next page)

(continued from previous page)

35	Diverse Steuerungen	AGFW Symposium DH	KA-0035	NTR_1_RL_Ein					
→ROHR	AUF	1	1.0	ROHR	R-1905	R-1906	AGFW Symposium		
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					
36	Diverse Steuerungen	AGFW Symposium DH	KA-0036	NTR_3_Aus_VL					
→ROHR	AUF	1	1.0	ROHR	V-3008	V-3007	AGFW Symposium		
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					
37	Diverse Steuerungen	AGFW Symposium DH	KA-0037	NTR_3_Aus_RL					
→ROHR	AUF	1	1.0	ROHR	R-3008	R-3007	AGFW Symposium		
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					
38	Diverse Steuerungen	AGFW Symposium DH	KA-0038	NTR_3_Ein_VL					
→ROHR	ZU	1	1.0	ROHR	V-3008	V-3007	AGFW Symposium		
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					
39	Diverse Steuerungen	AGFW Symposium DH	KA-0039	NTR_3_Ein_RL					
→ROHR	ZU	1	1.0	ROHR	R-3008	R-3007	AGFW Symposium		
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					
40	Diverse Steuerungen	AGFW Symposium DH	KA-0040	NTR_2_Aus_VL					
→ROHR	AUF	1	1.0	ROHR	V-1110	V-1111	AGFW Symposium		
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					
41	Diverse Steuerungen	AGFW Symposium DH	KA-0041	NTR_2_Aus_RL					
→ROHR	AUF	1	1.0	ROHR	R-1110	R-1111	AGFW Symposium		
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					
42	Diverse Steuerungen	AGFW Symposium DH	KA-0042	NTR_2_Ein_VL					
→ROHR	ZU	1	1.0	ROHR	V-1110	V-1111	AGFW Symposium		
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					
43	Diverse Steuerungen	AGFW Symposium DH	KA-0043	NTR_2_Ein_RL					
→ROHR	ZU	1	1.0	ROHR	R-1110	R-1111	AGFW Symposium		
→DH	NaN	NaN	NaN	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					
44	Diverse Steuerungen	AGFW Symposium DH	KA-0054						
→KNOT	PSOLL	1	NaN	NaN	NaN	NaN			
→NaN	NaN	NaN	A_DH_pDef	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					
45	Diverse Steuerungen	AGFW Symposium DH	KA-0055						
→KNOT	PSOLL	1	NaN	NaN	NaN	NaN			
→NaN	NaN	NaN	A_DH_pDef	NaN		NaN	NaN	NaN	
→NaN	NaN		NaN	NaN					

_vQVAR()

One row per Timeseries.

Returns:**columns****QVAR**

- NAME
- BESCHREIBUNG
- INTPOL

- ZEITOPTION

SERIES

- QM: 1st Value
- QM_min
- QM_max

QVAR ID

- pk

Raises: XmError

_vPVAR()

One row per Timeseries.

Returns:

columns**PVAR**

- NAME
- BESCHREIBUNG
- INTPOL
- ZEITOPTION

SERIES

- PH: 1st Value
- PH_min
- PH_max

QVAR ID

- pk

Raises: XmError

_vVKNO()

One row per Blocknode.

Returns:

columns

- NAME
- CONT (Blockname)
- fkKNOT
- fkCONT

Raises: XmError

_vKNOT (*vVKNO=None, vQVAR=None, vPVAR=None, vLFKT=None*)

One row per Node (KNOT).

Args:

- vVKNO

- vQVAR
- vPVAR
- vLFKT

Returns:

rows sequence: Xml

columns**KNOT**

- NAME
- BESCHREIBUNG
- IDREFERENZ

CONT

- CONT
- CONT_ID
- CONT_LFDNR
- CONT_VKNO (name of the Block/Container for NAME is a Blocknode)

KNOT

- KTYP
- LFAKT (Umrechnungsfaktor)
- QM_EIN

QVAR

- QVAR_NAME
- QM, QM_min, QM_max

LFKT

- LFKT_NAME
- LF, LF_min, LF_max

PVAR

- PVAR_NAME
- PH, PH_min, PH_max

Zugehoerigkeit

- PZON_NAME
- FSTF_NAME,STOF_NAME,GMIX_NAME
- UTMP_NAME

2L

- 2L_NAME
- 2L_KVR

KNOT

- KVR
- TE
- TM

KNOT

- XKOR, YKOR, ZKOR

KNOT IDs

- pk, tk

KNOT: plot-Coordinates

- pXCor: X-pXCorZero
- pYCor: Y-pYCorZero

Refs

- fkFQPS

Raises: XmError

vKNOTexpEBES (*AColIdx=0*)

Expands Resultcolumns in vKNOT: FWVB and ESQUELLSP.

Precondition:

- vFWVB added with Resultcolumns FWVB~*~*~*~W, FWVB~*~*~*~QM
- vKNOT added with Resultcolumn ESQUELLSP

Arguments:

- AColIdx: Idx (LFDNR-1) of the EG which represents qsA (and defines qsNotA)

new Cols:

- qsStr: ESQUELLSP of the node as plain text
- **qs_LFDNR_NAME: share of EBES Nr. LFDNR in the supply of the node - i.e.:**
 - qs_1_Drei Linden
 - qs_4_Biogas Wette
 - ...
- the columns are derived from column ESQUELLSP
- the column ESQUELLSP is unchanged
- qsSigStr
- qs100: Index beginnend mit 1 der EG die den jeweiligen Knoten zu 100% versorgt - 0 sonst (d.h. dieser Knoten wird von keiner EG zu 100% versorgt)
- qsSUM: Summe aller EGs - 100, wenn der Knoten zu 100% von den EGs versorgt wird
- qsA
- qsNotA
- qsAnzKnoten
- qsAnzFwvb
- qsKNOT~*~*~*~QM'

- `qsFWVB~*~*~*~W`
- `qsFWVB~*~*~*~QM`
- `qsigKNOT~*~*~*~QM'`
- `qsigFWVB~*~*~*~W`
- `qsigFWVB~*~*~*~QM`
- `qsRank`: `NrOfGroup`: order is perc. desc. in EBES LFDNR order (the ESQUELLSP with max. share of 1st EBES is No. 1)
- `qsRankAnzKnoten`: `NrOfGroup`: order is Anzahl Knoten (the ESQUELLSP with max. `NOFN`-odes is No. 1)
- `qsRank`: die mit der größten neg. Abnahme ist die 1.
- `qsRankAnzFwvb`
- `qsRankFWVB~*~*~*~W`
- `qsigRank`: lexikalisch absteigend (d.h. die alles 1-Signatur ist die 1. und die alle 0-Signatur ist die letzte)
- `qsigRankAnzKnoten`: die mit den wenigsten Knoten ist die 1.
- `qsigRank`: die mit der größten neg. Abnahme ist die 1.
- `qsigRankAnzFwvb`
- `qsigRankFWVB~*~*~*~W`
- `qsigqsRankFWVB~*~*~*~W`: lfd. Nr. QS nach W-Größe sig, qs absteigend (d.h. No. 1 ist W-größte QSPK in der W-größten QSIG)

Returns:

- `vKNOT` with the new Cols

Raises: `XmError`

```

>>> # -q -m 0 -s vKNOTExpEBES -y no -z no -w DHNetwork
>>> # ---
>>> xm=xms['DHNetwork']
>>> # ---
>>> vKNOT=xm.dataFrames['vKNOT']
>>> vKNOTExp=xm.vKNOTExpEBES()
>>> r,c=vKNOT.shape
>>> r2,c2=vKNOTExp.shape
>>> r==r2
True
>>> a=sorted(vKNOTExp['qsigStr'].unique()) # sortierte Liste der voneinander
↳ verschiedenen Quellsignaturen
>>> a
['000', '001', '010', '011', '100', '110']
>>> len(a)
6
>>> vKNOTExp['qsRank'].max() # Anzahl verschiedener Quellspektren unter
↳ Berücksichtigung von KVR / Numerierung EBES
26
>>> # KVR:
>>> # in Wärmenetzen haben RL-Knoten i.d.R. 0 0 0 ... für alle RL-Knoten
>>> # dieses QS wird voneinander verschieden gezählt von einem möglichen QS
↳ 0 0 0 ... für VL-Knoten (wenn es welche gibt die von keiner aktiven EG
↳ versorgt werden)

```

(continues on next page)

(continued from previous page)

```

>>> vKNOTexp['qsRankAnzKnoten'].max() # Anzahl verschiedener Quellspektren_
↳ unter Berücksichtigung von KVR / Numerierung Anzahl Knoten
26
>>> import pandas as pd
>>> pd.set_option('display.width', 333)
>>> print(vKNOTexp[['KVR', 'qs_1_A', 'qs_2_B', 'qs_3_C', 'qsigStr', 'qsAnzKnoten',
↳ 'qsRank', 'qsRankAnzKnoten']].drop_duplicates(keep='first').sort_values(by=[
↳ 'qsRank'])).to_string(index=False))

```

KVR	qs_1_A	qs_2_B	qs_3_C	qsigStr	qsAnzKnoten	qsRank	qsRankAnzKnoten
1	100	0	0	100	210	1	1
1	99	1	0	110	9	2	10
1	98	2	0	110	2	3	17
1	97	3	0	110	55	4	4
1	96	4	0	110	28	5	7
1	95	5	0	110	2	6	18
1	82	18	0	110	18	7	8
1	76	24	0	110	1	8	21
1	74	26	0	110	79	9	3
1	67	33	0	110	4	10	12
1	61	39	0	110	3	11	14
1	58	42	0	110	4	12	13
1	48	52	0	110	2	13	19
1	39	61	0	110	2	14	20
1	36	64	0	110	5	15	11
1	35	65	0	110	3	16	15
1	24	76	0	110	46	17	5
1	11	89	0	110	1	18	22
1	10	90	0	110	12	19	9
1	6	94	0	110	1	20	23
1	0	100	0	010	195	21	2
1	0	92	8	011	1	22	24
1	0	67	33	011	3	23	16
1	0	44	56	011	1	24	25
1	0	0	100	001	40	25	6
2	0	0	0	000	735	26	26

```

>>> print(vKNOTexp[['KVR', 'qs_1_A', 'qs_2_B', 'qs_3_C', 'qsAnzKnoten', 'qsRank',
↳ 'qsRankAnzKnoten']].drop_duplicates(keep='first').sort_values(by=[
↳ 'qsRankAnzKnoten'])).to_string(index=False))

```

KVR	qs_1_A	qs_2_B	qs_3_C	qsAnzKnoten	qsRank	qsRankAnzKnoten
1	100	0	0	210	1	1
1	0	100	0	195	21	2
1	74	26	0	79	9	3
1	97	3	0	55	4	4
1	24	76	0	46	17	5
1	0	0	100	40	25	6
1	96	4	0	28	5	7
1	82	18	0	18	7	8
1	10	90	0	12	19	9
1	99	1	0	9	2	10
1	36	64	0	5	15	11
1	67	33	0	4	10	12
1	58	42	0	4	12	13
1	61	39	0	3	11	14
1	35	65	0	3	16	15
1	0	67	33	3	23	16
1	98	2	0	2	3	17
1	95	5	0	2	6	18

(continues on next page)

(continued from previous page)

```

1      48      52      0      2      13      19
1      39      61      0      2      14      20
1      76      24      0      1       8      21
1      11      89      0      1      18      22
1       6      94      0      1      20      23
1       0      92      8      1      22      24
1       0      44     56      1      24      25
2       0       0       0     735     26      26
>>> print(vKNOTexp[['KVR','qs_1_A','qs_2_B','qs_3_C', 'qsAnzFwvb',
↳ 'qsRankFWVB~*~*~*~W','qsFWVB~*~*~*~W', 'qsFWVB~*~*~*~QM']].drop_
↳ duplicates(keep='first').sort_values(by=['qsRankFWVB~*~*~*~W']).round(-2).
↳ to_string(index=False))
KVR qs_1_A qs_2_B qs_3_C  qsAnzFwvb  qsRankFWVB~*~*~*~W  qsFWVB~*~*~*~W
↳ qsFWVB~*~*~*~QM
1      100      0      0      500.0      0      182000.0
↳ 2100.0
1       0     100      0      300.0      0      92500.0
↳ 1100.0
1      97       3      0      200.0      0      40100.0
↳ 500.0
1      74      26      0      200.0      0      34500.0
↳ 400.0
1      24      76      0      100.0      0      27400.0
↳ 300.0
1       0       0     100     100.0      0      25200.0
↳ 300.0
1      96       4      0     100.0      0      20500.0
↳ 200.0
1      11      89      0       0.0      0      18800.0
↳ 200.0
1      82      18      0       0.0      0      10900.0
↳ 100.0
1      95       5      0       0.0      0       6500.0
↳ 100.0
1      99       1      0       0.0      0       4700.0
↳ 100.0
1      58      42      0       0.0      0       4100.0
↳ 0.0
1      10      90      0       0.0      0       3800.0
↳ 0.0
1      36      64      0       0.0      0       2600.0
↳ 0.0
1      39      61      0       0.0      0       2500.0
↳ 0.0
1      35      65      0       0.0      0       2300.0
↳ 0.0
1      61      39      0       0.0      0       2100.0
↳ 0.0
1      67      33      0       0.0      0       1900.0
↳ 0.0
1       0      67      33       0.0      0       1300.0
↳ 0.0
1       6      94      0       0.0      0        900.0
↳ 0.0
1      98       2      0       0.0      0        600.0
↳ 0.0
1       0      44     56       0.0      0        500.0
↳ 0.0

```

(continues on next page)

(continued from previous page)

```

1      48      52      0      0.0      0      400.0
→      0.0
1      0      92      8      0.0      0      400.0
→      0.0
1      76      24      0      0.0      0      300.0
→      0.0
2      0      0      0      0.0      0      0.0
→      0.0
>>> # --- AnzFwvb in Knot
>>> vFWVB=xm.dataFrames['vFWVB']
>>> r,c=vFWVB.shape
>>> r==vKNOTexp['AnzFwvb'].sum()
True
>>> # --- Summe W in Knot
>>> WSumme=round(vFWVB['FWVB~*~*~*~W'].sum(),2)
>>> WSumme==round(vKNOTexp['FWVB~*~*~*~W'].sum(),2)
True
>>> # -----
>>> # in vKNOTexp sind die FWVB in Summe Anz und Summe W korrekt
>>> # -----
>>> grpObj=vKNOTexp[['KVR','KNOT~*~*~*~ESQUELLSP','FWVB~*~*~*~W','qsFWVB~*~*~
→*~W']].groupby(by=['KVR','KNOT~*~*~*~ESQUELLSP'],as_index=False)
>>> df=grpObj['FWVB~*~*~*~W'].sum().round(-2)
>>> df[['KVR','FWVB~*~*~*~W']]
   KVR  FWVB~*~*~*~W
0     1          25200.0
1     1           500.0
2     1          1300.0
3     1           400.0
4     1          92500.0
5     1           900.0
6     1          3800.0
7     1          18800.0
8     1          27400.0
9     1           2300.0
10    1          2600.0
11    1          2500.0
12    1           400.0
13    1          4100.0
14    1          2100.0
15    1          1900.0
16    1          34500.0
17    1           300.0
18    1          10900.0
19    1           6500.0
20    1          20500.0
21    1          40100.0
22    1           600.0
23    1          4700.0
24    1          182000.0
25    2              0.0
>>> round(WSumme,-3)==round(df['FWVB~*~*~*~W'].sum(),-3)
True
>>> # round(WSumme,0)
>>> df=grpObj['qsFWVB~*~*~*~W'].first()
>>> dfFirst=df[['KVR','qsFWVB~*~*~*~W']].round(1)
>>> df=grpObj['qsFWVB~*~*~*~W'].last()

```

(continues on next page)

(continued from previous page)

```

>>> dfLast=df[['KVR','qsFWVB~*~*~*~W']].round(1)
>>> # dfFirst
>>> # dfLast
>>> dfFirst.equals(dfLast)
True
>>> df=grpObj['qsFWVB~*~*~*~W'].mean()
>>> dfMean=df[['KVR','qsFWVB~*~*~*~W']].round(1)
>>> dfFirst.equals(dfMean)
True
>>> print(dfFirst.sort_values(by=['qsFWVB~*~*~*~W'],ascending=False).round(-
    ↪2).to_string(index=False))
KVR  qsFWVB~*~*~*~W
1      182000.0
1      92500.0
1      40100.0
1      34500.0
1      27400.0
1      25200.0
1      20500.0
1      18800.0
1      10900.0
1       6500.0
1       4700.0
1       4100.0
1       3800.0
1       2600.0
1       2500.0
1       2300.0
1       2100.0
1       1900.0
1       1300.0
1        900.0
1        600.0
1        500.0
1        400.0
1        400.0
1        300.0
2         0.0
>>> df=vKNOTexp[['KVR','qsigStr','qsigRankFWVB~*~*~*~W','qsigFWVB~*~*~*~W',
    ↪'qsigFWVB~*~*~*~QM']].drop_duplicates(keep='first').sort_values(by=[
    ↪'qsigRankFWVB~*~*~*~W']).round({'qsigFWVB~*~*~*~W': -2, 'qsigFWVB~*~*~*~QM
    ↪': -2})
>>> print(df.to_string(index=False))
KVR qsigStr  qsigRankFWVB~*~*~*~W  qsigFWVB~*~*~*~W  qsigFWVB~*~*~*~QM
1    110      1      184800.0      2200.0
1    100      2      182000.0      2100.0
1    010      3       92500.0      1100.0
1    001      4       25200.0       300.0
1    011      5        2200.0         0.0
2    000      6         0.0         0.0
>>> round(WSumme,-3)==round(df['qsigFWVB~*~*~*~W'].sum(),-3)
True
>>> grpObj=vKNOTexp.groupby(by=['qsigRankFWVB~*~*~*~W','qsRankFWVB~*~*~*~W'],
    ↪as_index=False)
>>> d={col:'min' for col in ['qsigStr','qs_1_A','qs_2_B','qs_3_C',
    ↪'qsigqsRankFWVB~*~*~*~W']}
>>> d.update({'qsigFWVB~*~*~*~W':'min'})

```

(continues on next page)

(continued from previous page)

```

>>> d.update({'qsFWVB~*~*~*~W': 'min'})
>>> d.update({'pk': 'count'}) # Anzahl Knoten
>>> d.update({'NAME': 'first'}) # ein Knotenname
>>> # d.update({'AnzFwvb': 'sum'}) # muss gleich d.update({'qsAnzFwvb': 'first
→'}) sein
>>> d.update({'qsigAnzFwvb': 'first'})
>>> d.update({'qsAnzFwvb': 'first'})
>>> df=grpObj.agg(d).sort_values(by=['qsigRankFWVB~*~*~*~W', 'qsRankFWVB~*~*~
→*~W'], ascending=True)
>>> df.rename(columns={'pk': 'AnzKnoten', 'NAME': 'NAME'}, inplace=True)
>>> xm.dataFrames['df']=df.round({'qsigFWVB~*~*~*~W': -2, 'qsFWVB~*~*~*~W': -
→2})
>>> print(xm._getvXXXXAsOneString(vXXXX='df'))
  qsigRankFWVB~*~*~*~W  qsRankFWVB~*~*~*~W  qsigStr  qs_1_A  qs_2_B  qs_3_C
→qsigqsRankFWVB~*~*~*~W  qsigFWVB~*~*~*~W  qsFWVB~*~*~*~W  AnzKnoten  1
→NAME  qsigAnzFwvb  qsAnzFwvb
0
→      1      3      110      97      3      0
→      1      184800.0      40100.0      55  V-1852
→      598.0      157.0
1
→      1      4      110      74      26      0
→      2      184800.0      34500.0      79  V-3611
→      598.0      150.0
2
→      1      5      110      24      76      0
→      3      184800.0      27400.0      46  V-1630
→      598.0      92.0
3
→      1      7      110      96      4      0
→      4      184800.0      20500.0      28  V-1773
→      598.0      58.0
4
→      1      8      110      11      89      0
→      5      184800.0      18800.0      1  V-3109
→      598.0      3.0
5
→      1      9      110      82      18      0
→      6      184800.0      10900.0      18  V-1712
→      598.0      38.0
6
→      1      10      110      95      5      0
→      7      184800.0      6500.0      2  V-1132
→      598.0      1.0
7
→      1      11      110      99      1      0
→      8      184800.0      4700.0      9  V-1335
→      598.0      17.0
8
→      1      12      110      58      42      0
→      9      184800.0      4100.0      4  V-1751
→      598.0      12.0
9
→      1      13      110      10      90      0
→      10      184800.0      3800.0      12  V-3426
→      598.0      19.0
10
→      1      14      110      36      64      0
→      11      184800.0      2600.0      5  V-1755
→      598.0      9.0
11
→      1      15      110      39      61      0
→      12      184800.0      2500.0      2  V-1372
→      598.0      4.0
12
→      1      16      110      35      65      0
→      13      184800.0      2300.0      3  V-1744
→      598.0      9.0
13
→      1      17      110      61      39      0
→      14      184800.0      2100.0      3  V-1742
→      598.0      7.0

```

(continues on next page)

(continued from previous page)

14		1		18	110	67	33	0	
→		15	184800.0		1900.0		4	V-1607	
→	598.0	8.0							
15		1		20	110	6	94	0	
→		16	184800.0		900.0		1	V-1374	
→	598.0	3.0							
16		1		21	110	98	2	0	
→		17	184800.0		600.0		2	V-1803	
→	598.0	7.0							
17		1		23	110	48	52	0	
→		18	184800.0		400.0		2	V-1308	
→	598.0	2.0							
18		1		25	110	76	24	0	
→		19	184800.0		300.0		1	V-1743	
→	598.0	2.0							
19		2		1	100	100	0	0	
→		20	182000.0		182000.0		210	V-1208	
→	485.0	485.0							
20		3		2	010	0	100	0	
→		21	92500.0		92500.0		195	V-3202	
→	338.0	338.0							
21		4		6	001	0	0	100	
→		22	25200.0		25200.0		40	V-2400	
→	74.0	74.0							
22		5		19	011	0	67	33	
→		23	2200.0		1300.0		3	V-2351	
→	10.0	5.0							
23		5		22	011	0	44	56	
→		24	2200.0		500.0		1	V-2352	
→	10.0	2.0							
24		5		24	011	0	92	8	
→		25	2200.0		400.0		1	V-2140	
→	10.0	3.0							
25		6		26	000	0	0	0	
→		26	0.0		0.0		735	R-3709	
→	0.0	0.0							

```

>>> import re
>>> qsColsEgr=[col for col in vKNOTExp.columns.tolist() if re.search('^qs_',
→col) != None]
>>> qsColsEgr
['qs_1_A', 'qs_2_B', 'qs_3_C']
>>> qsColsInf=[col for col in vKNOTExp.columns.tolist() if re.search('^qs',
→col) != None and re.search('^qs_',col) == None]
>>> qsColsInf
['qsStr', 'qsigStr', 'qs100', 'qsSUM', 'qsA', 'qsNotA', 'qsARank',
→'qsAnzKnoten', 'qsAnzFwvb', 'qsFWVB~~~~~W', 'qsFWVB~~~~~QM', 'qsRank',
→'qsRankAnzKnoten', 'qsRankAnzFwvb', 'qsRankFWVB~~~~~W', 'qsigAnzKnoten',
→'qsigAnzFwvb', 'qsigFWVB~~~~~W', 'qsigFWVB~~~~~QM', 'qsigRank',
→'qsigRankAnzKnoten', 'qsigRankAnzFwvb', 'qsigRankFWVB~~~~~W',
→'qsigqsRankFWVB~~~~~W']
>>> #xm=xms['OneLPipe']
>>> # ---
>>> #vKNOTExp=xm.vKNOTExpEBES()

```

vROHRExpEBES (vKNOTExpEBES)

Expands Resultcolumns in vROHR with ESQUELLSP-columns.

Arguments:

- vKNOTexpEBES: df with ESQUELLSP-columns

new Cols:

Returns:

- vROHR with the new Cols

Raises: `XmError`

```
>>> # -q -m 0 -s vROHReXP_EBES -y no -z no -w DHNetwork
>>> # ---
>>> import pandas as pd
>>> pd.set_option('display.width', 333)
>>> xm=xms['DHNetwork']
>>> vKNOTexp=xm.vKNOTexpEBES()
>>> vROHReXP=xm.vROHReXP_EBES(vKNOTexp)
>>> r,c=xm.dataFrames['vROHR'].shape
>>> r2,c2=vROHReXP.shape
>>> r==r2
True
>>> c2>c
True
>>> grpObj=vROHReXP.groupby(by=['qsigRankFWVB~*~*~*~W', 'qsRankFWVB~*~*~*~W'],
    ↳as_index=False)
>>> d={col:'min' for col in ['qsigStr', 'qs_1_A', 'qs_2_B', 'qs_3_C',
    ↳'qsigqsRankFWVB~*~*~*~W']}
>>> d.update({'qsigFWVB~*~*~*~W':'min'})
>>> d.update({'qsFWVB~*~*~*~W':'min'})
>>> #d.update({'NAME_k':'count'}) # Anzahl Knoten
>>> d.update({'NAME_i':'first'}) # ein Knotenname
>>> # d.update({'AnzFwvb':'sum'}) # muss gleich d.update({'qsAnzFwvb':'first
    ↳'}) sein
>>> d.update({'qsigAnzFwvb':'first'})
>>> d.update({'qsAnzFwvb':'first'})
>>> d.update({'pk':'count'}) # Anzahl Rohre
>>> #d.update({'L':'sum'}) # Länge Rohre
>>> d.update({'qsigRank_sumL':'first'})
>>> d.update({'qsRank_sumL':'first'})
>>> d.update({'qsigRank_L':'first'})
>>> d.update({'qsRank_L':'first'})
>>> df=grpObj.agg(d).sort_values(by=['qsigRankFWVB~*~*~*~W', 'qsRankFWVB~*~*~
    ↳~*~W'], ascending=True)
>>> df.rename(columns={'NAME_k':'AnzKnoten', 'NAME_i':'1 NAME', 'pk':'AnzRohre
    ↳'}, inplace=True)
>>> xm.dataFrames['df']=df[['qsigRankFWVB~*~*~*~W', 'qsRankFWVB~*~*~*~W',
    ↳'qsigStr', 'qs_1_A', 'qs_2_B', 'qs_3_C', 'qsigqsRankFWVB~*~*~*~W', 'qsigFWVB~*~
    ↳~*~W', 'qsFWVB~*~*~*~W', '1 NAME', 'qsigAnzFwvb', 'qsAnzFwvb', 'AnzRohre',
    ↳'qsigRank_sumL']].round({'qsigRankFWVB~*~*~*~W':-2, 'qsRankFWVB~*~*~*~W':-2,
    ↳'qsigqsRankFWVB~*~*~*~W':-2, 'qsigFWVB~*~*~*~W':-2, 'qsFWVB~*~*~*~W':-2,
    ↳'qsigRank_sumL':-2})
>>> print(xm._getvXXXXAsOneString(vXXXX='df'))
    qsigRankFWVB~*~*~*~W  qsRankFWVB~*~*~*~W qsigStr qs_1_A qs_2_B qs_3_C
    ↳qsigqsRankFWVB~*~*~*~W  qsigFWVB~*~*~*~W  qsFWVB~*~*~*~W  1 NAME
    ↳qsigAnzFwvb  qsAnzFwvb  AnzRohre  qsigRank_sumL
0                0                0      110      97        3        0
    ↳
    ↳0      157.0        72      51200.0      40100.0  V-1802      598.
1                0                0      110      74        26        0
    ↳
    ↳0                0      184800.0      34500.0  V-3505      598.
    ↳0      150.0        89      51200.0
```

(continues on next page)

(continued from previous page)

2		0		0	110	24	76	0	┐
→		0	184800.0		27400.0	V-1633		598.	
→0	92.0	53	51200.0						
3		0		0	110	96	4	0	┐
→		0	184800.0		20500.0	V-1114		598.	
→0	58.0	33	51200.0						
4		0		0	110	82	18	0	┐
→		0	184800.0		10900.0	V-1711		598.	
→0	38.0	20	51200.0						
5		0		0	110	95	5	0	┐
→		0	184800.0		6500.0	V-1132		598.	
→0	1.0	1	51200.0						
6		0		0	110	99	1	0	┐
→		0	184800.0		4700.0	V-1336		598.	
→0	17.0	8	51200.0						
7		0		0	110	58	42	0	┐
→		0	184800.0		4100.0	V-1750		598.	
→0	12.0	4	51200.0						
8		0		0	110	10	90	0	┐
→		0	184800.0		3800.0	V-3420		598.	
→0	19.0	11	51200.0						
9		0		0	110	36	64	0	┐
→		0	184800.0		2600.0	V-1755		598.	
→0	9.0	5	51200.0						
10		0		0	110	39	61	0	┐
→		0	184800.0		2500.0	V-1373		598.	
→0	4.0	2	51200.0						
11		0		0	110	35	65	0	┐
→		0	184800.0		2300.0	V-1743		598.	
→0	9.0	5	51200.0						
12		0		0	110	61	39	0	┐
→		0	184800.0		2100.0	V-1740		598.	
→0	7.0	3	51200.0						
13		0		0	110	67	33	0	┐
→		0	184800.0		1900.0	V-1605		598.	
→0	8.0	4	51200.0						
14		0		0	110	6	94	0	┐
→		0	184800.0		900.0	V-1374		598.	
→0	3.0	1	51200.0						
15		0		0	110	98	2	0	┐
→		0	184800.0		600.0	V-1802		598.	
→0	7.0	3	51200.0						
16		0		0	110	48	52	0	┐
→		0	184800.0		400.0	V-1308		598.	
→0	2.0	3	51200.0						
17		0		0	110	76	24	0	┐
→		0	184800.0		300.0	V-1742		598.	
→0	2.0	1	51200.0						
18		0		0	100	100	0	0	┐
→		0	182000.0		182000.0	V-1591		485.	
→0	485.0	261	36800.0						
19		0		0	010	0	100	0	┐
→		0	92500.0		92500.0	V-3204		338.	
→0	338.0	214	37400.0						
20		0		0	001	0	0	100	┐
→		0	25200.0		25200.0	V-2602		74.	
→0	74.0	43	7300.0						

(continues on next page)

(continued from previous page)

```

21          0          0    011    0    67    33
↳          0          2200.0    1300.0 V-2113    10.
↳0      5.0          3          300.0
22          0          0    000    0    0    0
↳          0          0.0          0.0 R-1226    0.
↳0      0.0      839    133000.0
>>> grpObj=vROHRexp.groupby(by=['qsigRank_L','qsRank_L'],as_index=False)
>>> d={col:'min' for col in ['qsigStr','qs_1_A','qs_2_B','qs_3_C',
↳'qsigqsRank_L']}
>>> d.update({'qsigFWVB~*~*~*~W':'min'})
>>> d.update({'qsFWVB~*~*~*~W':'min'})
>>> #d.update({'NAME_k':'count'}) # Anzahl Knoten
>>> d.update({'NAME_i':'first'}) # ein Knotenname
>>> # d.update({'AnzFwvb':'sum'}) # muss gleich d.update({'qsAnzFwvb':'first
↳'}) sein
>>> d.update({'qsigAnzFwvb':'first'})
>>> d.update({'qsAnzFwvb':'first'})
>>> d.update({'pk':'count'}) # Anzahl Rohre
>>> #d.update({'L':'sum'}) # Länge Rohre
>>> d.update({'qsigRank_sumL':'first'})
>>> d.update({'qsRank_sumL':'first'})
>>> #d.update({'qsigRank_L':'first'})
>>> #d.update({'qsRank_L':'first'})
>>> df=grpObj.agg(d).sort_values(by=['qsigRank_L','qsRank_L'],ascending=True)
>>> df.rename(columns={'NAME_k':'AnzKnoten','NAME_i':'1 NAME','pk':'AnzRohre
↳'},inplace=True)
>>> xm.dataFrames['df']=df.round({'qsigFWVB~*~*~*~W':-2,'qsFWVB~*~*~*~W':-2,
↳'qsigRank_sumL':-2,'qsRank_sumL':-2})
>>> print(xm._getvXXXXAsOneString(vXXXX='df'))
    qsigRank_L  qsRank_L qsigStr qs_1_A qs_2_B qs_3_C  qsigqsRank_L
↳qsigFWVB~*~*~*~W  qsFWVB~*~*~*~W  1 NAME  qsigAnzFwvb  qsAnzFwvb  AnzRohre
↳ qsigRank_sumL  qsRank_sumL
0          1          3    110    74    26    0          1
↳184800.0      34500.0 V-3505      598.0    150.0      89
↳51200.0      16200.0
1          1          4    110    97    3    0          2
↳184800.0      40100.0 V-1802      598.0    157.0      72
↳51200.0      9800.0
2          1          5    110    24    76    0          3
↳184800.0      27400.0 V-1633      598.0     92.0      53
↳51200.0      9700.0
3          1          7    110    96    4    0          4
↳184800.0      20500.0 V-1114      598.0     58.0      33
↳51200.0      4500.0
4          1          8    110    82    18    0          5
↳184800.0      10900.0 V-1711      598.0     38.0      20
↳51200.0      2900.0
5          1          9    110    10    90    0          6
↳184800.0      3800.0 V-3420      598.0     19.0      11
↳51200.0      2200.0
6          1         10    110    99    1    0          7
↳184800.0      4700.0 V-1336      598.0     17.0      8
↳51200.0      1400.0
7          1         11    110    35    65    0          8
↳184800.0      2300.0 V-1743      598.0     9.0      5
↳51200.0      800.0
8          1         12    110    48    52    0          9
↳184800.0      400.0 V-1308      598.0     2.0
↳51200.0      600.0

```

(continues on next page)

(continued from previous page)

9	1	13	110	36	64	0	10	
↪184800.0		2600.0	V-1755		598.0	9.0	5	
↪51200.0		600.0						
10	1	14	110	67	33	0	11	
↪184800.0		1900.0	V-1605		598.0	8.0	4	
↪51200.0		600.0						
11	1	15	110	98	2	0	12	
↪184800.0		600.0	V-1802		598.0	7.0	3	
↪51200.0		400.0						
12	1	16	110	58	42	0	13	
↪184800.0		4100.0	V-1750		598.0	12.0	4	
↪51200.0		400.0						
13	1	17	110	6	94	0	14	
↪184800.0		900.0	V-1374		598.0	3.0	1	
↪51200.0		400.0						
14	1	18	110	61	39	0	15	
↪184800.0		2100.0	V-1740		598.0	7.0	3	
↪51200.0		300.0						
15	1	19	110	39	61	0	16	
↪184800.0		2500.0	V-1373		598.0	4.0	2	
↪51200.0		300.0						
16	1	21	110	95	5	0	17	
↪184800.0		6500.0	V-1132		598.0	1.0	1	
↪51200.0		200.0						
17	1	22	110	76	24	0	18	
↪184800.0		300.0	V-1742		598.0	2.0	1	
↪51200.0		100.0						
18	2	1	010	0	100	0	19	
↪ 92500.0		92500.0	V-3204		338.0	338.0	214	
↪37400.0		37400.0						
19	3	2	100	100	0	0	20	
↪182000.0		182000.0	V-1591		485.0	485.0	261	
↪36800.0		36800.0						
20	4	6	001	0	0	100	21	
↪ 25200.0		25200.0	V-2602		74.0	74.0	43	
↪7300.0		7300.0						
21	5	20	011	0	67	33	22	
↪ 2200.0		1300.0	V-2113		10.0	5.0	3	
↪ 300.0		300.0						
22	6	23	000	0	0	0	23	
↪ 0.0		0.0	R-1226		0.0	0.0	839	
↪133000.0		133000.0						

_vROHR (*vKNOT=None*)

One row per Pipe (ROHR).

Args: vKNOT**Returns:****rows** sequence: Xml**columns****ROHR**

- BESCHREIBUNG
- IDREFERENZ

ROHR

- BAUJAHR, HAL
- IPLANUNG, KENNUNG

ROHR

- L, LZU, RAU, ZAUS, ZEIN, ZUML
- JLAMBS, LAMBDA0

ROHR

- ASOLL, INDSCHALL

ROHR FW

- NAME_i_2L
- NAME_k_2L
- KVR

DTRO_ROW

- AUSFALLZEIT, DA , DI , DN , KT , PN , REHABILITATION , REPARATUR , S , WSTEIG , WTIEFE

LTGR

- LTGR_NAME, LTGR_BESCHREIBUNG , SICHTBARKEIT , VERLEGEART

DTRO

- DTRO_NAME, DTRO_BESCHREIBUNG, E

REF

- fkSTRASSE, fkSRAT

ROHR IDs

- pk, tk

ROHR BZ

- ITRENN
- LECKSTART, LECKEND, LECKMENGE, LECKORT, LECKSTATUS

Rest

- QSVB, ZVLIMPTNZ, KANTENZV

CONT

- CONT
- CONT_ID
- CONT_LFDNR

vKNOT

KI

- NAME_i
- KVR_i, TM_i

- XKOR_i, YKOR_i, ZKOR_i

KK

- NAME_k
- KVR_k, TM_k
- XKOR_k, YKOR_k, ZKOR_k

pXCor_i, pYCor_i # X / Y des KNOTens i pXCor_k, pYCor_k # X / Y des KNOTens k

PLOT

- pXCors, pYCors # KNOTenkoordinaten i,k als je 2-elementige Liste
- pWAYPXCors, pWAYPYCors # um min. X / min. Y aller Knoten der Netzansicht bereinigte Wegpunktkoordinatenlisten, d.h. der Wegpunkt “ganz links unten” hat die Koordinaten 0/0

Raises: XmError

_vFWVB (*vKNOT=None*, *vLFKT=None*, *vWBLZ=None*)
One row per DistrictHeatingHousestation (FWVB).

Args:

- vKNOT
- vLFKT
- wWBLZ

Returns:**columns****FWVB**

- BESCHREIBUNG
- IDREFERENZ
- W0
- LFK
- W0LFK
- TVL0, TRS0

vLFKT

- LFKT
- W, W_min, W_max

FWVB contd.

- INDTR, TRSK
- VTYP
- DPHAUS, IMBG, IRFV

FWVB IDs

- pk, tk

vKNOT

Ki

- NAME_i
- KVR_i, TM_i
- XKOR_i, YKOR_i, ZKOR_i
- pXCor_i, pYCor_i

Kk

- NAME_k
- KVR_k, TM_i
- XKOR_k, YKOR_k, ZKOR_i
- pXCor_k, pYCor_i

vCONT

- CONT
- CONT_ID
- CONT_LFDNR

vWBLZ

- ['BLZ1','BLZ2',...] list of the WBLZ-Names of the FWVB in alphabetical Order; empty list, if FWVB is not a WBLZ-Member

Raises: XmError

_vVBEL (*vKNOT=None, edges=['ROHR', 'VENT', 'FWVB', 'FWES', 'PUMP', 'KLAP', 'REGV', 'PREG', 'MREG', 'DPRG', 'PGRP'], edgesD=['', 'DN', '', 'DN', '', 'DN', 'DN', 'DN', 'DN', 'DN', ''], mColNames=['OBJTYPE', 'pk'], mIdxNames=['OBJTYPE', 'OBJID']*)
One row per Edge.

Args:

- vKNOT: df
- edges: list of strs
- edgesD: list of strs
- mColNames: list of columns which shall be used as MIndex; the columns will be dropped; the columns must be delivered by _vVBEL_XXXX
- mIdxNames: list of names for the indices for the columns above

Returns: Edge-df returned Edge-df is None if an exception occurs

rows:

- sequence edges: edges
- sequence within edges: Xml

Mindices:

- OBJTYPE: str: 'ROHR','VENT',... [default a MIndex not a column]
- OBJID [default a MIndex not a column]

columns:

- LAYR

- L in m (0 if edge <> ROHR)
- D in mm (NaN if no Diameter could be determined)

columns:

- see _vVBEL_XXXX

_vVBEL_XXXX (*vKNOT=None, OBJTYPE=None*)

One row per Edge.

Args:

- vKNOT: df
- **OBJTYPE**: str ('ROHR','VENT',...) self.dataFrames[OBJTYPE] is used to build with vKNOT the returned Edge-df

Returns: Edge-df None is returned if an exception occurs

columns:

- OBJTYPE: str: ROHR,VENT,...
- BESCHREIBUNG
- IDREFERENZ
- pk
- tk
- NAME_i
- CONT_i
- CONT_VKNO_i
- Z_i
- pk_i
- NAME_k
- CONT_k
- CONT_VKNO_k
- Z_k
- pk_k

_vRXXX (*nodes=['RSLW', 'RMES', 'RHYS', 'RLVG', 'RLSR', 'RMMA', 'RADD', 'RMUL', 'RDIV', 'RTOT', 'RPT1', 'RINT', 'RPID', 'RFKT', 'RSTN']*)

One row per R-Node.

Args:

- nodes: List of all R-Node Elements but RUES

Returns: R-Node df returned R-Node df is None if an exception occurs

rows:

- sequence nodes: nodes
- sequence within nodes: Xml

columns:

- see _vRXXX_XXXX

```

>>> import pandas as pd
>>> pd.set_option('display.width', 333)
>>> # ---
>>> xm=xms['DHNetwork']
>>> # ---
>>> vRXXX=xm.dataFrames['vRXXX']
>>> vRXXX['lfdNr']=range(len(vRXXX))
>>> vRXXX

```

	OBJTYPE	BESCHREIBUNG	KA	CONT
→	pk	tk lfdNr		
0	RSLW	Leck_1_Ein Leck_1_Ein	AGFW Symposium DH	↵
→	5706111677806224290	5706111677806224290	0	
1	RSLW	Leck_2_Ein Leck_2_Ein	AGFW Symposium DH	↵
→	4704869532416514405	4704869532416514405	1	
2	RSLW	Leck_3_Ein Leck_3_Ein	AGFW Symposium DH	↵
→	4808434710442736644	4808434710442736644	2	
3	RSLW	wNA wNA	AGFW Symposium DH	↵
→	4991855568438544033	4991855568438544033	3	
4	RSLW	wNB wNB	AGFW Symposium DH	↵
→	4658075570394029953	4658075570394029953	4	
5	RSLW	wNC wNC	AGFW Symposium DH	↵
→	5240575308071562858	5240575308071562858	5	
6	RSLW	vorOrtNA vorOrtNA	AGFW Symposium DH	↵
→	5194343043762135519	5194343043762135519	6	
7	RSLW	vorOrtNB vorOrtNB	AGFW Symposium DH	↵
→	4705080808435797677	4705080808435797677	7	
8	RSLW	vorOrtNC vorOrtNC	AGFW Symposium DH	↵
→	5620348872583735825	5620348872583735825	8	
9	RSLW	wLast wLast	AGFW Symposium DH	↵
→	5741660563170722352	5741660563170722352	9	
10	RSLW	wTRST wTRST	AGFW Symposium DH	↵
→	5547011912763631199	5547011912763631199	10	
11	RSLW	Leck_Menge Leck_Menge	AGFW Symposium DH	↵
→	5390061625789905096	5390061625789905096	11	
12	RSLW	Leck_VL Leck_VL	AGFW Symposium DH	↵
→	4880440884169110259	4880440884169110259	12	
13	RSLW	Leck_RL Leck_RL	AGFW Symposium DH	↵
→	5644481773793849108	5644481773793849108	13	
14	RSLW	wDH_RD_A wDH_RD_A	AGFW Symposium DH	↵
→	4622192786925004485	4622192786925004485	14	
15	RSLW	wDH_MD_A wDH_MD_A	AGFW Symposium DH	↵
→	5093705160009582980	5093705160009582980	15	
16	RSLW wDH_BA_A; 1=RD; 0=MD	wDH_BA_A_RD	AGFW Symposium DH	↵
→	5322890886142492590	5322890886142492590	16	
17	RSLW	dpA dpA	AGFW Symposium DH	↵
→	4849866990207957614	4849866990207957614	17	
18	RSLW	qB qB	AGFW Symposium DH	↵
→	4771725364091629759	4771725364091629759	18	
19	RSLW	qC qC	AGFW Symposium DH	↵
→	4978409087288292434	4978409087288292434	19	
20	RSLW	None 0	Diverse Steuerungen	↵
→	5486870913514090048	5486870913514090048	20	
21	RSLW	None 1	Diverse Steuerungen	↵
→	5377084992102722959	5377084992102722959	21	
22	RSLW	Analog Dummy ADum	Diverse Steuerungen	↵
→	5408457159782566744	5408457159782566744	22	
23	RSLW	None 100	Diverse Steuerungen	↵
→	5055797784689898209	5055797784689898209	23	

(continues on next page)

(continued from previous page)

24	RSLW	NaN	cp	Sekundärwerte	↪
↪4838608935279518502		4838608935279518502	24		
0	RMES	NaN	yUWM	Sekundärwerte	↪
↪5008805081156446169		5008805081156446169	25		
1	RMES	NaN	mP	Sekundärwerte	↪
↪5180980864512333141		5180980864512333141	26		
2	RMES	NaN	TRSP	Sekundärwerte	↪
↪4964809001779537631		4964809001779537631	27		
3	RMES	NaN	TVL	Sekundärwerte	↪
↪5137355888694407298		5137355888694407298	28		
4	RMES	NaN	wLastMW	Sekundärwerte	↪
↪4833634373103605497		4833634373103605497	29		
5	RMES	NaN	yLastMW	Sekundärwerte	↪
↪4817923247686815456		4817923247686815456	30		
6	RMES	NaN	yAMW	Sekundärwerte	↪
↪4726758453134789052		4726758453134789052	31		
7	RMES	NaN	yBMW	Sekundärwerte	↪
↪5528896084200811302		5528896084200811302	32		
8	RMES	NaN	yCMW	Sekundärwerte	↪
↪5274276049082272588		5274276049082272588	33		
9	RMES	NaN	dUWMMin	Sekundärwerte	↪
↪5463544828758888616		5463544828758888616	34		
10	RMES	NaN	dUWMMMax	Sekundärwerte	↪
↪4672771372882677276		4672771372882677276	35		
11	RMES	NaN	KA-0026	Sekundärwerte	↪
↪5714273708462554381		5714273708462554381	36		
12	RMES	NaN	QDHGes	Sekundärwerte	↪
↪5345716897595312355		5345716897595312355	37		
13	RMES	dp / 2	yDH_dp2_A	Sekundärwerte	↪
↪5512879293670562022		5512879293670562022	38		
14	RMES	None	yDH_pRL_A	Sekundärwerte	↪
↪4639451967914783278		4639451967914783278	39		
0	RLVG	NaN	wNAEin_vO	AGFW Symposium DH	↪
↪4742316320267545359		4742316320267545359	40		
1	RLVG	NaN	wNBEin_vO	AGFW Symposium DH	↪
↪5013654033692161674		5013654033692161674	41		
2	RLVG	NaN	wNCEin_vO	AGFW Symposium DH	↪
↪5670691593026035398		5670691593026035398	42		
3	RLVG	NaN	wDH_BA_A_MD	AGFW Symposium DH	↪
↪4873987359791313088		4873987359791313088	43		
4	RLVG	NaN	Leck_1_VL	Diverse Steuerungen	↪
↪5669152199869266879		5669152199869266879	44		
5	RLVG	NaN	nLeck_1_VL	Diverse Steuerungen	↪
↪5517055963660007188		5517055963660007188	45		
6	RLVG	NaN	KA-0001	Diverse Steuerungen	↪
↪4937005671108174325		4937005671108174325	46		
7	RLVG	NaN	KA-0002	Diverse Steuerungen	↪
↪5752519230439786595		5752519230439786595	47		
8	RLVG	NaN	KA-0009	Diverse Steuerungen	↪
↪5660961189098354654		5660961189098354654	48		
9	RLVG	NaN	KA-0010	Diverse Steuerungen	↪
↪5510085446018401887		5510085446018401887	49		
10	RLVG	NaN	KA-0011	Diverse Steuerungen	↪
↪4894802981639605379		4894802981639605379	50		
11	RLVG	NaN	KA-0012	Diverse Steuerungen	↪
↪5310832758005678867		5310832758005678867	51		
12	RLVG	NaN	KA-0017	Diverse Steuerungen	↪
↪4879781051055847299		4879781051055847299	52		

(continues on next page)

(continued from previous page)

13	RLVG	NaN	KA-0018	Diverse Steuerungen	↪
↪	4806239740367977881	4806239740367977881	53		
14	RLVG	NaN	KA-0019	Diverse Steuerungen	↪
↪	5447964234902471608	5447964234902471608	54		
15	RLVG	NaN	KA-0020	Diverse Steuerungen	↪
↪	4717907439365620025	4717907439365620025	55		
0	RADD	NaN	dT	Sekundärwerte	↪
↪	4654077245127093202	4654077245127093202	56		
1	RADD	NaN	dLastMW	Sekundärwerte	↪
↪	4611793887272861500	4611793887272861500	57		
2	RADD	NaN	yUWMLast	Sekundärwerte	↪
↪	5574611204646558662	5574611204646558662	58		
3	RADD	NaN	yDH_pMD_A	Sekundärwerte	↪
↪	5255402486218254174	5255402486218254174	59		
4	RADD	NaN	wDH_MD_A_ERO	Diverse Steuerungen	↪
↪	5729434727271745948	5729434727271745948	60		
0	RSTN	NaN	wNA_RSTN	A	↪
↪	5165635044767172069	5165635044767172069	61		
1	RSTN	NaN	KA-0046	A	↪
↪	5137384799783014264	5137384799783014264	62		
2	RSTN	NaN	KA-0044	A	↪
↪	5636962607360173089	5636962607360173089	63		
3	RSTN	NaN	KA-0045	A	↪
↪	5597572325891198144	5597572325891198144	64		
4	RSTN	NaN	wNB_RSTN	B	↪
↪	5342104608381486733	5342104608381486733	65		
5	RSTN	NaN	KA-0053	B	↪
↪	5338620382667478180	5338620382667478180	66		
6	RSTN	NaN	KA-0057	B	↪
↪	5226612456739754122	5226612456739754122	67		
7	RSTN	NaN	KA-0058	B	↪
↪	5537037692802520861	5537037692802520861	68		
8	RSTN	NaN	wNC_RSTN	C	↪
↪	5103693862180601916	5103693862180601916	69		
9	RSTN	NaN	KA-0059	C	↪
↪	4792266770335818241	4792266770335818241	70		
10	RSTN	NaN	KA-0060	C	↪
↪	5286169822203128424	5286169822203128424	71		
11	RSTN	NaN	KA-0061	C	↪
↪	4848495011382561496	4848495011382561496	72		
12	RSTN	NaN	KA-0004	Diverse Steuerungen	↪
↪	5625633953643797107	5625633953643797107	73		
13	RSTN	NaN	KA-0005	Diverse Steuerungen	↪
↪	4851348857631426312	4851348857631426312	74		
14	RSTN	NaN	KA-0006	Diverse Steuerungen	↪
↪	5185169121447805605	5185169121447805605	75		
15	RSTN	NaN	KA-0008	Diverse Steuerungen	↪
↪	4760680402451575539	4760680402451575539	76		
16	RSTN	NaN	KA-0003	Diverse Steuerungen	↪
↪	5249070009027066113	5249070009027066113	77		
17	RSTN	NaN	KA-0007	Diverse Steuerungen	↪
↪	5721409231684230901	5721409231684230901	78		
18	RSTN	NaN	KA-0013	Diverse Steuerungen	↪
↪	5075554822852863012	5075554822852863012	79		
19	RSTN	NaN	KA-0014	Diverse Steuerungen	↪
↪	5320878233009751638	5320878233009751638	80		
20	RSTN	NaN	KA-0015	Diverse Steuerungen	↪
↪	5749069735826810904	5749069735826810904	81		

(continues on next page)

(continued from previous page)

21	RSTN	NaN	KA-0016	Diverse Steuerungen	↪
↪	5704472379299329003	5704472379299329003	82		
22	RSTN	NaN	KA-0021	Diverse Steuerungen	↪
↪	5629658054546932585	5629658054546932585	83		
23	RSTN	NaN	KA-0022	Diverse Steuerungen	↪
↪	5162821695312832398	5162821695312832398	84		
24	RSTN	NaN	KA-0023	Diverse Steuerungen	↪
↪	5357357577779591773	5357357577779591773	85		
25	RSTN	NaN	KA-0024	Diverse Steuerungen	↪
↪	5357348958190741976	5357348958190741976	86		
26	RSTN	NaN	KA-0025	Diverse Steuerungen	↪
↪	4635966862484721732	4635966862484721732	87		
27	RSTN	NaN	KA-0027	Diverse Steuerungen	↪
↪	5700600513951468434	5700600513951468434	88		
28	RSTN	NaN	KA-0028	Diverse Steuerungen	↪
↪	5367185280774605989	5367185280774605989	89		
29	RSTN	NaN	KA-0029	Diverse Steuerungen	↪
↪	5445770133105602710	5445770133105602710	90		
30	RSTN	NaN	KA-0030	Diverse Steuerungen	↪
↪	4885570100974274375	4885570100974274375	91		
31	RSTN	NaN	KA-0031	Diverse Steuerungen	↪
↪	5223383850171539514	5223383850171539514	92		
32	RSTN	NTR_1_RL_Ein	KA-0032	Diverse Steuerungen	↪
↪	5333724089944967011	5333724089944967011	93		
33	RSTN	NTR_1_VL_Ein	KA-0033	Diverse Steuerungen	↪
↪	4825143842549434339	4825143842549434339	94		
34	RSTN	NTR_1_VL_Ein	KA-0034	Diverse Steuerungen	↪
↪	4693599139501858956	4693599139501858956	95		
35	RSTN	NTR_1_RL_Ein	KA-0035	Diverse Steuerungen	↪
↪	5693503518255620080	5693503518255620080	96		
36	RSTN	NTR_3_Aus_VL	KA-0036	Diverse Steuerungen	↪
↪	4901609871029871596	4901609871029871596	97		
37	RSTN	NTR_3_Aus_RL	KA-0037	Diverse Steuerungen	↪
↪	5327767233627106399	5327767233627106399	98		
38	RSTN	NTR_3_Ein_VL	KA-0038	Diverse Steuerungen	↪
↪	5760765754619184144	5760765754619184144	99		
39	RSTN	NTR_3_Ein_RL	KA-0039	Diverse Steuerungen	↪
↪	4994799657516451637	4994799657516451637	100		
40	RSTN	NTR_2_Aus_VL	KA-0040	Diverse Steuerungen	↪
↪	5348583181653286363	5348583181653286363	101		
41	RSTN	NTR_2_Aus_RL	KA-0041	Diverse Steuerungen	↪
↪	5499083775210733192	5499083775210733192	102		
42	RSTN	NTR_2_Ein_VL	KA-0042	Diverse Steuerungen	↪
↪	4990389026836623226	4990389026836623226	103		
43	RSTN	NTR_2_Ein_RL	KA-0043	Diverse Steuerungen	↪
↪	5697786347617919077	5697786347617919077	104		
44	RSTN	NaN	KA-0054	Diverse Steuerungen	↪
↪	5623872434691357889	5623872434691357889	105		
45	RSTN	NaN	KA-0055	Diverse Steuerungen	↪
↪	5283008774827895454	5283008774827895454	106		

_vRXXX_XXXX (*OBJTYPE=None*)

One row per R-Node of Type OBJTYPE.

Args: OBJTYPE: str: i.e. RHYS**Returns:** R-Node df None is returned if an exception occurs

columns:

- OBJTYPE: str: i.e. RADD
- BESCHREIBUNG
- KA
- CONT
- pk
- tk

MxSync (*mx=None, ForceNoH5ReadForMx=False, ForceNoH5Update=False*)

Xm: NEW 1st Call: vNRCV_Mx1: vNRCV with MX1-Information. Some Xm-Views with MX2-Information (mx2Idx).Mx: Sir3sID Update in Mx-Object.

Args:**mx (default: None): Mx-Object**

- If no Mx-Object is given the Mx-Object corresponding to the Xm-Object is constructed and returned.
- **Notes:**
 - **The Sync-Result in Xm is persisted if xm were read from H5:**
 - * xm.ToH5() is called if xm.h5Read is True and not ForceNoH5Update.
 - **The Sync-Result in Mx is persisted if mx were read from H5:**
 - * mx.ToH5() is called (from __Mx1_Sir3sIDUpd) if Sir3sID-Updates occurred and mx.h5Read is True and not ForceNoH5Update.

ForceNoH5ReadForMx (default: False): has an Effect only if a new Mx-Object is constructed

- **ForceNoH5ReadForMx = True:**
 - the new Mx-Object is constructed with NoH5Read=True
- **ForceNoH5ReadForMx = False:**
 - the new Mx-Object is constructed with NoH5Read = not self.h5Read
 - if the Xm was read from H5 the Mx is constructed with NoH5Read=False
 - if the Xm was not read from H5 the Mx is constructed with NoH5Read=True

ForceNoH5Update (default: False): if read from H5, H5 is updated if ForceNoH5Update is False

Returns: Mx-Object if no Mx-Object was given; Nothing else

Raises: XmlError

```
>>> # -q -m 0 -s MxAdd -t both -y yes -z no -w LocalHeatingNetwork
>>> xm=xms['LocalHeatingNetwork']
>>> xm.h5Read # False due to MockUp
False
>>> mx=xm.MxSync()
>>> mx.h5Read # False due to MockUp
False
```

`_vREdges()`

Returns vREdges.

columns:

OBJTYPE_Ki BESCHREIBUNG_Ki Kn_Ki: Node ID pk_Ki: Node ID

OBJTYPE_Kk BESCHREIBUNG_Kk Kn_Kk pk_Kk CONT

KnExt_Ki: with OBJTYPE extended Node ID KnExt_Kk: with OBJTYPE extended Node ID

```
>>> import pandas as pd
>>> # ---
>>> xm=xms['DHNetwork']
>>> # ---
>>> vREdges=xm.dataFrames['vREdges']
>>> pd.set_option('display.width', 333)
>>> pd.set_option('display.max_columns', None)
>>> pd.set_option('display.max_rows', None)
>>> vREdges[[
...   'CONT'
...   , 'CONT_PARENT'
...   , 'OBJTYPE_Ki'
...   , 'OBJTYPE_Kk'
...   , 'Kn_Ki'
...   , 'Kn_Kk'
...   , 'KnExt_Ki'
...   , 'KnExt_Kk'
... ]].sort_values(by=['KnExt_Ki', 'KnExt_Kk', 'CONT']).sort_index()
```

	CONT	CONT_PARENT	OBJTYPE_Ki	OBJTYPE_Kk	
→Kn_Ki	Kn_Kk	KnExt_Ki	KnExt_Kk		
0	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	Leck_
→1_Ein	Leck_1_Ein	Leck_1_Ein_RSLW	Leck_1_Ein_RUES		
1	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	Leck_
→2_Ein	Leck_2_Ein	Leck_2_Ein_RSLW	Leck_2_Ein_RUES		
2	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	Leck_
→3_Ein	Leck_3_Ein	Leck_3_Ein_RSLW	Leck_3_Ein_RUES		
3	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	
→ wNA	wNA	wNA_RSLW	wNA_RUES		
4	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	
→ wNB	wNB	wNB_RSLW	wNB_RUES		
5	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	
→ wNC	wNC	wNC_RSLW	wNC_RUES		
6	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	
→vorOrtNA	vorOrtNA	vorOrtNA_RSLW	vorOrtNA_RUES		
7	AGFW Symposium DH	AGFW Symposium DH	RUES	RLVG	
→vorOrtNA	wNAEin_vO	vorOrtNA_RUES	wNAEin_vO_RLVG		
8	AGFW Symposium DH	AGFW Symposium DH	RLVG	RUES	
→wNAEin_vO	wNAEin	wNAEin_vO_RLVG	wNAEin_RUES		
9	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	
→vorOrtNB	vorOrtNB	vorOrtNB_RSLW	vorOrtNB_RUES		
10	AGFW Symposium DH	AGFW Symposium DH	RUES	RLVG	
→vorOrtNB	wNBEin_vO	vorOrtNB_RUES	wNBEin_vO_RLVG		
11	AGFW Symposium DH	AGFW Symposium DH	RLVG	RUES	
→wNBEin_vO	wNBEin	wNBEin_vO_RLVG	wNBEin_RUES		
12	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	
→vorOrtNC	vorOrtNC	vorOrtNC_RSLW	vorOrtNC_RUES		
13	AGFW Symposium DH	AGFW Symposium DH	RUES	RLVG	
→vorOrtNC	wNCEin_vO	vorOrtNC_RUES	wNCEin_vO_RLVG		

(continues on next page)

(continued from previous page)

14	AGFW Symposium DH	AGFW Symposium DH	RLVG	RUES	↪
↪wNCEin_vO	wNCEin	wNCEin_vO_RLVG	wNCEin_RUES		
15	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	↪
↪wLast	wLast	wLast_RSLW	wLast_RUES		
16	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	↪
↪wTRST	wTRST	wTRST_RSLW	wTRST_RUES		
17	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	Leck_↪
↪Menge	Leck_Menge	Leck_Menge_RSLW	Leck_Menge_RUES		
18	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	↪
↪Leck_VL	Leck_VL	Leck_VL_RSLW	Leck_VL_RUES		
19	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	↪
↪Leck_RL	Leck_RL	Leck_RL_RSLW	Leck_RL_RUES		
20	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	wDH_↪
↪RD_A	wDH_RD_A	wDH_RD_A_RSLW	wDH_RD_A_RUES		
21	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	wDH_↪
↪MD_A	wDH_MD_A	wDH_MD_A_RSLW	wDH_MD_A_RUES		
22	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	wDH_BA_↪
↪A_RD	wDH_BA_A_RD	wDH_BA_A_RD_RSLW	wDH_BA_A_RD_RUES		
23	AGFW Symposium DH	AGFW Symposium DH	RUES	RLVG	wDH_BA_↪
↪A_RD	wDH_BA_A_MD	wDH_BA_A_RD_RUES	wDH_BA_A_MD_RLVG		
24	AGFW Symposium DH	AGFW Symposium DH	RLVG	RUES	wDH_BA_↪
↪A_MD	wDH_BA_A_MD	wDH_BA_A_MD_RLVG	wDH_BA_A_MD_RUES		
25	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	↪
↪dpA	dpA	dpA_RSLW	dpA_RUES		
26	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	↪
↪qB	qB	qB_RSLW	qB_RUES		
27	AGFW Symposium DH	AGFW Symposium DH	RSLW	RUES	↪
↪qC	qC	qC_RSLW	qC_RUES		
28	Diverse Steuerungen	AGFW Symposium DH	RSLW	RUES	↪
↪0	0	0_RSLW	0_RUES		
29	Diverse Steuerungen	AGFW Symposium DH	RSLW	RUES	↪
↪1	1	1_RSLW	1_RUES		
30	Diverse Steuerungen	AGFW Symposium DH	RSLW	RUES	↪
↪ADum	ADum	ADum_RSLW	ADum_RUES		
31	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	↪
↪wLast	KA-0004	wLast_RUES	KA-0004_RSTN		
32	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	↪
↪1	KA-0004	1_RUES	KA-0004_RSTN		
33	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	↪
↪wTRST	KA-0005	wTRST_RUES	KA-0005_RSTN		
34	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	↪
↪1	KA-0005	1_RUES	KA-0005_RSTN		
35	Diverse Steuerungen	AGFW Symposium DH	RSLW	RUES	↪
↪100	100	100_RSLW	100_RUES		
36	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	Leck_↪
↪1_Ein	Leck_1_VL	Leck_1_Ein_RUES	Leck_1_VL_RLVG		
37	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	↪
↪Leck_VL	Leck_1_VL	Leck_VL_RUES	Leck_1_VL_RLVG		
38	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	Leck_↪
↪1_VL	KA-0006	Leck_1_VL_RLVG	KA-0006_RSTN		
39	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	↪
↪ADum	KA-0006	ADum_RUES	KA-0006_RSTN		
40	Diverse Steuerungen	AGFW Symposium DH	RLVG	RLVG	Leck_↪
↪1_VL	nLeck_1_VL	Leck_1_VL_RLVG	nLeck_1_VL_RLVG		
41	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	↪
↪ADum	KA-0008	ADum_RUES	KA-0008_RSTN		
42	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	nLeck_↪
↪1_VL	KA-0008	nLeck_1_VL_RLVG	KA-0008_RSTN		

(continues on next page)

(continued from previous page)

43	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	Leck_
→1_Ein	KA-0001	Leck_1_Ein_RUES	KA-0001_RLVG		
44	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	└
→Leck_RL	KA-0001	Leck_RL_RUES	KA-0001_RLVG		
45	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	KA-
→0001	KA-0003	KA-0001_RLVG	KA-0003_RSTN		
46	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	└
→ADum	KA-0003	ADum_RUES	KA-0003_RSTN		
47	Diverse Steuerungen	AGFW Symposium DH	RLVG	RLVG	KA-
→0001	KA-0002	KA-0001_RLVG	KA-0002_RLVG		
48	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	└
→ADum	KA-0007	ADum_RUES	KA-0007_RSTN		
49	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	KA-
→0002	KA-0007	KA-0002_RLVG	KA-0007_RSTN		
50	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	└
→Leck_RL	KA-0011	Leck_RL_RUES	KA-0011_RLVG		
51	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	Leck_
→2_Ein	KA-0011	Leck_2_Ein_RUES	KA-0011_RLVG		
52	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	Leck_
→2_Ein	KA-0009	Leck_2_Ein_RUES	KA-0009_RLVG		
53	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	└
→Leck_VL	KA-0009	Leck_VL_RUES	KA-0009_RLVG		
54	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	KA-
→0009	KA-0013	KA-0009_RLVG	KA-0013_RSTN		
55	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	└
→ADum	KA-0013	ADum_RUES	KA-0013_RSTN		
56	Diverse Steuerungen	AGFW Symposium DH	RLVG	RLVG	KA-
→0009	KA-0010	KA-0009_RLVG	KA-0010_RLVG		
57	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	└
→ADum	KA-0015	ADum_RUES	KA-0015_RSTN		
58	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	KA-
→0010	KA-0015	KA-0010_RLVG	KA-0015_RSTN		
59	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	KA-
→0011	KA-0014	KA-0011_RLVG	KA-0014_RSTN		
60	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	└
→ADum	KA-0014	ADum_RUES	KA-0014_RSTN		
61	Diverse Steuerungen	AGFW Symposium DH	RLVG	RLVG	KA-
→0011	KA-0012	KA-0011_RLVG	KA-0012_RLVG		
62	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	└
→ADum	KA-0016	ADum_RUES	KA-0016_RSTN		
63	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	KA-
→0012	KA-0016	KA-0012_RLVG	KA-0016_RSTN		
64	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	└
→Leck_RL	KA-0019	Leck_RL_RUES	KA-0019_RLVG		
65	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	Leck_
→3_Ein	KA-0019	Leck_3_Ein_RUES	KA-0019_RLVG		
66	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	Leck_
→3_Ein	KA-0017	Leck_3_Ein_RUES	KA-0017_RLVG		
67	Diverse Steuerungen	AGFW Symposium DH	RUES	RLVG	└
→Leck_VL	KA-0017	Leck_VL_RUES	KA-0017_RLVG		
68	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	KA-
→0017	KA-0021	KA-0017_RLVG	KA-0021_RSTN		
69	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	└
→ADum	KA-0021	ADum_RUES	KA-0021_RSTN		
70	Diverse Steuerungen	AGFW Symposium DH	RLVG	RLVG	KA-
→0017	KA-0018	KA-0017_RLVG	KA-0018_RLVG		
71	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	└
→ADum	KA-0023	ADum_RUES	KA-0023_RSTN		

(continues on next page)

(continued from previous page)

72	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	KA-
→0018	KA-0023	KA-0018_RLVG	KA-0023_RSTN		
73	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	KA-
→0019	KA-0022	KA-0019_RLVG	KA-0022_RSTN		
74	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→ADum	KA-0022	ADum_RUES	KA-0022_RSTN		
75	Diverse Steuerungen	AGFW Symposium DH	RLVG	RLVG	KA-
→0019	KA-0020	KA-0019_RLVG	KA-0020_RLVG		
76	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→ADum	KA-0024	ADum_RUES	KA-0024_RSTN		
77	Diverse Steuerungen	AGFW Symposium DH	RLVG	RSTN	KA-
→0020	KA-0024	KA-0020_RLVG	KA-0024_RSTN		
78	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	Leck_
→Menge	KA-0025	Leck_Menge_RUES	KA-0025_RSTN		
79	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→1	KA-0025	1_RUES	KA-0025_RSTN		
80	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	Leck_
→Menge	KA-0027	Leck_Menge_RUES	KA-0027_RSTN		
81	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→1	KA-0027	1_RUES	KA-0027_RSTN		
82	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	Leck_
→Menge	KA-0028	Leck_Menge_RUES	KA-0028_RSTN		
83	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→1	KA-0028	1_RUES	KA-0028_RSTN		
84	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	Leck_
→Menge	KA-0029	Leck_Menge_RUES	KA-0029_RSTN		
85	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→1	KA-0029	1_RUES	KA-0029_RSTN		
86	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	Leck_
→Menge	KA-0030	Leck_Menge_RUES	KA-0030_RSTN		
87	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→1	KA-0030	1_RUES	KA-0030_RSTN		
88	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	Leck_
→Menge	KA-0031	Leck_Menge_RUES	KA-0031_RSTN		
89	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→1	KA-0031	1_RUES	KA-0031_RSTN		
90	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→ADum	KA-0032	ADum_RUES	KA-0032_RSTN		
91	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→0	KA-0032	0_RUES	KA-0032_RSTN		
92	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→0	KA-0033	0_RUES	KA-0033_RSTN		
93	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→ADum	KA-0033	ADum_RUES	KA-0033_RSTN		
94	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→0	KA-0034	0_RUES	KA-0034_RSTN		
95	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→ADum	KA-0034	ADum_RUES	KA-0034_RSTN		
96	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→0	KA-0035	0_RUES	KA-0035_RSTN		
97	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→ADum	KA-0035	ADum_RUES	KA-0035_RSTN		
98	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→0	KA-0036	0_RUES	KA-0036_RSTN		
99	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→ADum	KA-0036	ADum_RUES	KA-0036_RSTN		
100	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	┐
→0	KA-0037	0_RUES	KA-0037_RSTN		

(continues on next page)

(continued from previous page)

101	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ADum	KA-0037	ADum_RUES	KA-0037_RSTN		
102	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ 0	KA-0038	0_RUES	KA-0038_RSTN		
103	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ADum	KA-0038	ADum_RUES	KA-0038_RSTN		
104	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ 0	KA-0039	0_RUES	KA-0039_RSTN		
105	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ADum	KA-0039	ADum_RUES	KA-0039_RSTN		
106	Diverse Steuerungen	AGFW Symposium DH	RUES	RADD	yDH_
→dp2_A	wDH_MD_A_ERO	yDH_dp2_A_RUES	wDH_MD_A_ERO_RADD		
107	Diverse Steuerungen	AGFW Symposium DH	RUES	RADD	wDH_
→MD_A	wDH_MD_A_ERO	wDH_MD_A_RUES	wDH_MD_A_ERO_RADD		
108	Diverse Steuerungen	AGFW Symposium DH	RADD	RUES	wDH_MD_
→A_ERO	wDH_MD_A_ERO	wDH_MD_A_ERO_RADD	wDH_MD_A_ERO_RUES		
109	Diverse Steuerungen	AGFW Symposium DH	RUES	RUES	wDH_
→RD_A	wDH_RD_A_ERO	wDH_RD_A_RUES	wDH_RD_A_ERO_RUES		
110	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	wDH_RD_
→A_ERO	KA-0054	wDH_RD_A_ERO_RUES	KA-0054_RSTN		
111	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	wDH_BA_
→A_RD	KA-0054	wDH_BA_A_RD_RUES	KA-0054_RSTN		
112	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	wDH_MD_
→A_ERO	KA-0055	wDH_MD_A_ERO_RUES	KA-0055_RSTN		
113	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	wDH_BA_
→A_MD	KA-0055	wDH_BA_A_MD_RUES	KA-0055_RSTN		
114	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ 0	KA-0040	0_RUES	KA-0040_RSTN		
115	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ADum	KA-0040	ADum_RUES	KA-0040_RSTN		
116	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ 0	KA-0041	0_RUES	KA-0041_RSTN		
117	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ADum	KA-0041	ADum_RUES	KA-0041_RSTN		
118	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ 0	KA-0042	0_RUES	KA-0042_RSTN		
119	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ADum	KA-0042	ADum_RUES	KA-0042_RSTN		
120	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ 0	KA-0043	0_RUES	KA-0043_RSTN		
121	Diverse Steuerungen	AGFW Symposium DH	RUES	RSTN	
→ADum	KA-0043	ADum_RUES	KA-0043_RSTN		
122		A	AGFW Symposium DH	RUES	RSTN
→wNAEin	wNA_RSTN	wNAEin_RUES	wNA_RSTN_RSTN		
123		A	AGFW Symposium DH	RUES	RSTN
→ wNA	wNA_RSTN	wNA_RUES	wNA_RSTN_RSTN		
124		A	AGFW Symposium DH	RUES	RSTN
→wNBEin	KA-0044	wNBEin_RUES	KA-0044_RSTN		
125		A	AGFW Symposium DH	RUES	RSTN
→ADum	KA-0044	ADum_RUES	KA-0044_RSTN		
126		A	AGFW Symposium DH	RUES	RSTN
→vorOrtNC	KA-0045	vorOrtNC_RUES	KA-0045_RSTN		
127		A	AGFW Symposium DH	RUES	RSTN
→ADum	KA-0045	ADum_RUES	KA-0045_RSTN		
128		A	AGFW Symposium DH	RUES	RSTN
→ dpA	KA-0046	dpA_RUES	KA-0046_RSTN		
129		A	AGFW Symposium DH	RUES	RSTN
→vorOrtNC	KA-0046	vorOrtNC_RUES	KA-0046_RSTN		

(continues on next page)

(continued from previous page)

130		B	AGFW Symposium DH	RUES	RSTN	
↪wNBEin	wNB_RSTN		wNBEin_RUES	wNB_RSTN_RSTN		┌
131		B	AGFW Symposium DH	RUES	RSTN	
↪ wNB	wNB_RSTN		wNB_RUES	wNB_RSTN_RSTN		┌
132		B	AGFW Symposium DH	RUES	RSTN	
↪wNBEin	KA-0053		wNBEin_RUES	KA-0053_RSTN		┌
133		B	AGFW Symposium DH	RUES	RSTN	
↪ADum	KA-0053		ADum_RUES	KA-0053_RSTN		┌
134		B	AGFW Symposium DH	RUES	RSTN	
↪vorOrtNB	KA-0057		vorOrtNB_RUES	KA-0057_RSTN		┌
135		B	AGFW Symposium DH	RUES	RSTN	
↪ADum	KA-0057		ADum_RUES	KA-0057_RSTN		┌
136		B	AGFW Symposium DH	RUES	RSTN	
↪ qB	KA-0058		qB_RUES	KA-0058_RSTN		┌
137		B	AGFW Symposium DH	RUES	RSTN	
↪vorOrtNB	KA-0058		vorOrtNB_RUES	KA-0058_RSTN		┌
138		C	AGFW Symposium DH	RUES	RSTN	
↪wNCEin	wNC_RSTN		wNCEin_RUES	wNC_RSTN_RSTN		┌
139		C	AGFW Symposium DH	RUES	RSTN	
↪ wNC	wNC_RSTN		wNC_RUES	wNC_RSTN_RSTN		┌
140		C	AGFW Symposium DH	RUES	RSTN	
↪wNCEin	KA-0059		wNCEin_RUES	KA-0059_RSTN		┌
141		C	AGFW Symposium DH	RUES	RSTN	
↪ADum	KA-0059		ADum_RUES	KA-0059_RSTN		┌
142		C	AGFW Symposium DH	RUES	RSTN	
↪ADum	KA-0060		ADum_RUES	KA-0060_RSTN		┌
143		C	AGFW Symposium DH	RUES	RSTN	
↪vorOrtNC	KA-0060		vorOrtNC_RUES	KA-0060_RSTN		┌
144		C	AGFW Symposium DH	RUES	RSTN	
↪ qC	KA-0061		qC_RUES	KA-0061_RSTN		┌
145		C	AGFW Symposium DH	RUES	RSTN	
↪vorOrtNC	KA-0061		vorOrtNC_RUES	KA-0061_RSTN		┌
146	Sekundärwerte		Diverse Steuerungen	RMES	RADD	
↪TRSP	dT		TRSP_RMES	dT_RADD		┌
147	Sekundärwerte		Diverse Steuerungen	RMES	RADD	
↪ TVL	dT		TVL_RMES	dT_RADD		┌
148	Sekundärwerte		Diverse Steuerungen	RMES	RUES	
↪yUWM	yUWM		yUWM_RMES	yUWM_RUES		┌
149	Sekundärwerte		Diverse Steuerungen	RMES	RUES	
↪wLastMW	wLastMW		wLastMW_RMES	wLastMW_RUES		┌
150	Sekundärwerte		Diverse Steuerungen	RMES	RUES	
↪yLastMW	yLastMW		yLastMW_RMES	yLastMW_RUES		┌
151	Sekundärwerte		Diverse Steuerungen	RUES	RADD	
↪yLastMW	dLastMW		yLastMW_RUES	dLastMW_RADD		┌
152	Sekundärwerte		Diverse Steuerungen	RUES	RADD	
↪wLastMW	dLastMW		wLastMW_RUES	dLastMW_RADD		┌
153	Sekundärwerte		Diverse Steuerungen	RADD	RUES	
↪dLastMW	dLastMW		dLastMW_RADD	dLastMW_RUES		┌
154	Sekundärwerte		Diverse Steuerungen	RMES	RUES	
↪yAMW	yAMW		yAMW_RMES	yAMW_RUES		┌
155	Sekundärwerte		Diverse Steuerungen	RMES	RUES	
↪yBMW	yBMW		yBMW_RMES	yBMW_RUES		┌
156	Sekundärwerte		Diverse Steuerungen	RMES	RUES	
↪yCMW	yCMW		yCMW_RMES	yCMW_RUES		┌
157	Sekundärwerte		Diverse Steuerungen	RMES	RUES	
↪dUWMMin	dUWMMin		dUWMMin_RMES	dUWMMin_RUES		┌
158	Sekundärwerte		Diverse Steuerungen	RMES	RUES	
↪dUWMMMax	dUWMMMax		dUWMMMax_RMES	dUWMMMax_RUES	(continues on next page)	┌

(continued from previous page)

```

159      Sekundärwerte   Diverse Steuerungen      RUES      RADD
→yUWM      yUWMLast      yUWM_RUES      yUWMLast_RADD
160      Sekundärwerte   Diverse Steuerungen      RMES      RADD      KA-
→0026      yUWMLast      KA-0026_RMES      yUWMLast_RADD
161      Sekundärwerte   Diverse Steuerungen      RADD      RUES
→yUWMLast      yUWMLast      yUWMLast_RADD      yUWMLast_RUES
162      Sekundärwerte   Diverse Steuerungen      RMES      RUES
→QDHGes      QDHGes      QDHGes_RMES      QDHGes_RUES
163      Sekundärwerte   Diverse Steuerungen      RMES      RUES      yDH_
→dp2_A      yDH_dp2_A      yDH_dp2_A_RMES      yDH_dp2_A_RUES
164      Sekundärwerte   Diverse Steuerungen      RUES      RADD      yDH_
→dp2_A      yDH_pMD_A      yDH_dp2_A_RUES      yDH_pMD_A_RADD
165      Sekundärwerte   Diverse Steuerungen      RUES      RADD      yDH_
→pRL_A      yDH_pMD_A      yDH_pRL_A_RUES      yDH_pMD_A_RADD
166      Sekundärwerte   Diverse Steuerungen      RADD      RUES      yDH_
→pMD_A      yDH_pMD_A      yDH_pMD_A_RADD      yDH_pMD_A_RUES
167      Sekundärwerte   Diverse Steuerungen      RMES      RUES      yDH_
→pRL_A      yDH_pRL_A      yDH_pRL_A_RMES      yDH_pRL_A_RUES
>>> import networkx as nx
>>> G=nx.from_pandas_edgelist(vREdges, source='KnExt_Ki', target='KnExt_Kk',
→edge_attr=True,create_using=nx.DiGraph())
>>> list(nx.selfloop_edges(G))
[]
>>> pathNodes=nx.shortest_path(G,'Leck_1_Ein_RSLW','KA-0008_RSTN')
>>> pathNodes
['Leck_1_Ein_RSLW', 'Leck_1_Ein_RUES', 'Leck_1_VL_RLVG', 'nLeck_1_VL_RLVG',
→'KA-0008_RSTN']
>>> sink_nodes = [node for node, outdegree in G.out_degree(G.nodes()) if
→outdegree == 0]
>>> source_nodes = [node for node, indegree in G.in_degree(G.nodes()) if
→indegree == 0]
>>> import re
>>> for source, sink in [(source, sink) for sink in sink_nodes for source in
→source_nodes]: # ueber alle Quellen pro Senke ...
...     if re.search('_RSTN$',sink) != None:
...         for path in nx.all_simple_paths(G, source=source, target=sink):
...             if sink=='KA-0008_RSTN':
...                 path
['Leck_1_Ein_RSLW', 'Leck_1_Ein_RUES', 'Leck_1_VL_RLVG', 'nLeck_1_VL_RLVG',
→'KA-0008_RSTN']
['Leck_VL_RSLW', 'Leck_VL_RUES', 'Leck_1_VL_RLVG', 'nLeck_1_VL_RLVG', 'KA-
→0008_RSTN']
['ADum_RSLW', 'ADum_RUES', 'KA-0008_RSTN']
>>> #---
>>> # dasselbe mit Knotennamen ohne Postfix ...
>>> G=nx.from_pandas_edgelist(vREdges, source='Kn_Ki', target='Kn_Kk', edge_
→attr=True,create_using=nx.DiGraph())
>>> # alle RUES Eingänge deren ID mit der des aufnehmenden Signals identisch
→ist führen dann zu Schleifen ...
>>> # ... die entfernt werden muessen wenn Quellen am Indegree erkannt
→werden sollen ...
>>> G.remove_edges_from(list(nx.selfloop_edges(G)))
>>> pathNodes=nx.shortest_path(G,'Leck_1_Ein','KA-0008')
>>> pathNodes # (auf die Pfadknotensequenz haben Schleifen keinen Einfluss,
→das Ergebnis waere mit den Schleifen dasselbe ...)
['Leck_1_Ein', 'Leck_1_VL', 'nLeck_1_VL', 'KA-0008']
>>> sink_nodes = [node for node, outdegree in G.out_degree(G.nodes()) if
→outdegree == 0]

```

(continues on next page)

(continued from previous page)

```

>>> source_nodes = [node for node, indegree in G.in_degree(G.nodes()) if
↳indegree == 0]
>>> for source, sink in [(source, sink) for sink in sink_nodes for source in
↳source_nodes]: # ueber alle Quellen pro Senke ...
...     for path in nx.all_simple_paths(G, source=source, target=sink):
...         if sink=='KA-0008':
...             path
['Leck_1_Ein', 'Leck_1_VL', 'nLeck_1_VL', 'KA-0008']
['Leck_VL', 'Leck_1_VL', 'nLeck_1_VL', 'KA-0008']
['ADum', 'KA-0008']

```

__MxSyncAddMx (*ForceNoH5ReadForMx=False*)

Mx-Object corresponding to the Xm-Object is constructed and returned.

ForceNoH5ReadForMx (default: False)

- **ForceNoH5ReadForMx = True:**
 - the new Mx-Object is constructed with NoH5Read=True
- **ForceNoH5ReadForMx = False:**
 - **the new Mx-Object is constructed with NoH5Read = not self.h5Read**
 - * if the Xm was read from H5 the Mx is constructed with NoH5Read=False
 - * if the Xm was not read from H5 the Mx is constructed with NoH5Read=True

Returns: Mx-Object

Raises: XmlError

__Mx1_Sir3sIDUpd (*mx, checkAllChannels=True, ForceNoH5Update=False*)

Update NAME1,2 and Sir3sID in mx.mx1Df and mx.df.

Args: mx: Mx-Object checkAllChannels: if False only Channels with empty NAME1 are updated; default: True: all Channels are checked and updated if necessary ForceNoH5Update: if False, H5 is updated if read from H5

Notes: Das Update ist erforderlich, weil SIR 3S die hier aktualisierten Kanal-Attribute *_nicht* nachführt, wenn diese sich im referenzierten Objekt geändert haben. Die Nachführung hier stellt sicher, dass der Sir3sID Kanalbezeichner, der sich aus Mx Kanal-Attributen ergibt dem Sir3sID Kanalbezeichner aus Xm Sachdaten-Attributen entspricht.

Unabhängig von der Nachführung: SirCalc: WARNUNG MXX Es gibt in der MX1-Datei ... Datenpunkte mit falschem DATATLENGTH-Attributwert WARNUNG MXX ... ungueltige oder unbekannte Datenpunkte erhalten das Ergebnis 0 oder Leerzeichen Das ist ein Hinweis auf Zombie-Kanäle, Kanäle deren Objekt nicht (mehr) existiert.

Diese Sachstände führen beim Nachführen zu ... Spalten mit demselben Namen (nan) in mx.df. Spalten mit demselben Namen sind generell in mehrfacher Hinsicht ungeeignet. Kommen sie vor, werden sie wie folgt umbenannt: Vorher: x,x,x,... Nachher: x,x_1,x_2, ...

The following OBJTYPEs are covered:

- KNOT
- WBLZ
- RXXXX

- alle Kanäle von Objekten die in vVBEL vorkommen

mx.ToH5() is called if Sir3sID-Updates occurred and mx.h5Read is True and not ForceNoH5Update.

Raises: XmError

```
>>> xm=xms['LocalHeatingNetwork']
>>> (wDir,modelDir,modelName,mx1File)=xm.getWDirModelDirModelName()
>>> try:
...     import Mx
... except:
...     from PT3S import Mx
>>> mx=Mx.Mx(mx1File=mx1File)
>>> # mx.mx1Df
>>> Sir3sIDStr='FWVB~V-K003~R-K003~5695730293103267172~INDUV'
>>> Sir3sIDStr='FWVB~~~5695730293103267172~INDUV'
>>> mx.mx1Df.loc[mx.mx1Df['Sir3sID']==Sir3sIDStr,'NAME1']='Sir3sIDUpdTest'
>>> mx.mx1Df.loc[mx.mx1Df['Sir3sID']==Sir3sIDStr,'Sir3sID']='FWVB~
↳Sir3sIDUpdTest~R-K003~5695730293103267172~INDUV'
>>> print(mx.mx1Df.loc[mx.mx1Df['NAME1']=='Sir3sIDUpdTest',['Sir3sID']].to_
↳string(index=False))

                                Sir3sID
FWVB~Sir3sIDUpdTest~R-K003~5695730293103267172~INDUV
>>> xm._Xm__Mx1_Sir3sIDUpd(mx)
>>> print(mx.mx1Df.loc[mx.mx1Df['Sir3sID']=='FWVB~V-K003~R-K003~
↳5695730293103267172~INDUV',['NAME1']].to_string(index=False))
NAME1
V-K003
>>> # -----
>>> # doppelte Spaltennamen behandeln
>>> # -----
>>> mx.df.rename(columns={'PUMP~R-1~R2~5481331875203087055~ETAW':
↳'Sir3sIDUpdTest'},inplace=True)
>>> mx.df.rename(columns={'PUMP~R-1~R2~5481331875203087055~DP':
↳'Sir3sIDUpdTest'},inplace=True)
>>> list(mx.df.columns[mx.df.columns.duplicated()])
['Sir3sIDUpdTest']
>>> mx.df.filter(regex='^Sir3sIDUpdTest').round(1).head(2)
                                Sir3sIDUpdTest  Sir3sIDUpdTest
2004-09-22 08:30:00+00:00                    0.6                2.3
2004-09-22 08:30:15+00:00                    0.7                1.3
>>> xm._Xm__Mx1_Sir3sIDUpd(mx)
>>> mx.df.filter(regex='^Sir3sIDUpdTest').round(1).head(2)
                                Sir3sIDUpdTest  Sir3sIDUpdTest_1
2004-09-22 08:30:00+00:00                    0.6                2.3
2004-09-22 08:30:15+00:00                    0.7                1.3
```

__Mx1_Sir3sIDUpd_ObjTypeNode (mx=None, dfUpd=None, dfNAME1=None, NAME1Col='NAME')

Update Sir3sID and NAME1 in mx.mx1Df and mx.df for Channels in dfUpd.

Args: mx: Mx-Object dfUpd: df with OBJTYPE,NAME1,NAME2,OBJTYPE_PK,ATTRTYPE,Sir3sID to be updated dfNAME1: df with NAME1-Information NAME1Col: col in dfNAME1 with NAME1-Information

Returns: nOfSir3sIDsUpdated

Note: only wrong Channels are updated

Raises: XmError

__Mx1_vNRCV (*mx*)
vNRCV_Mx1 (vNRCV with Mx1-Information) is added to dataFrames.

Args: mx: Mx-Object

self.dataFrames['vNRCV_Mx1']

index

- reindex

FILTERed

- existing MX-Channels only
- cRefLfdNr: 1st references only

SORTed

- Sir3sID

columns NEW

- Sir3sID

Raises: XmError

__Mx2_vROHR (*mx*)
Mx2-Information into vROHR.

Args: mx: Mx-Object

self.dataFrames['vROHR']

columns NEW

- mx2NoPts
- mx2Idx

Raises: XmError

__Mx2_vFWVB (*mx*)
Mx2-Information into vFWVB.

Args: mx: Mx-Object

self.dataFrames['vFWVB']

columns NEW

- mx2Idx

Raises: XmError

__Mx2_vKNOT (*mx*)
Mx2-Information into vKNOT.

Args: mx: Mx-Object

self.dataFrames['vKNOT']

columns NEW

- mx2Idx

Raises: XmError

__Mx2_vVBEL (*mx*, *edges*=['ROHR', 'VENT', 'FWVB', 'FWES', 'PUMP', 'KLAP', 'REGV', 'PREG', 'MREG', 'DPRG', 'PGRP'])
Mx2-Information into vVBEL.

Args: *mx*: Mx-Object

self.dataFrames['vVBEL']:

columns NEW

- *mx2Idx*
- **Notes:**
 - for all edges *mx2Idx* is taken from *mx.mx2Df*

Raises: *XmError*

MxAdd (*mx=None*, *ForceNoH5ReadForMx=False*, *ForceNoH5Update=False*, *timeReq=None*, *aggReq=None*, *timeReq2nd=None*, *viewList=[]*)
Add MX-Resultcolumn-Set to some Xm-Views. A Result-Set from previous Calls is deleted. NEW 1st Call: *vROHRVecResults*, *vAGSN*.

Args:

- *mx*, *ForceNoH5ReadForMx*, *ForceNoH5Update* : same Args as for *MxSync*; see description there
- **timeReq:**
 - **TIMESTAMP** (defining the MX-Resultcolumn-Set)
 - if *None* 1st TIME in *Mx* is used
 - if *aggReq* considered as **TIMESTAMPL**
- **aggReq (defining the MX-Resultcolumn-Set):**
 - 'TIME', 'TMIN', 'TMAX' (source: *MXS*) or 'MIN', 'MAX', ... (source: *mx.getVecAggs()*)
 - if not *None*, *timeReq* und *timeReq2nd* define the timespan
 - **if List**
 - * MX-Resultcolumns for several times/timespans are calculated
 - * *timeReq* and *timeReq2nd* must also be Lists
 - * if *viewList* is not *None*, in the views in *viewList* several MX-Resultcolumn-Sets are added: one per requested time/timespan
 - * the 2nd Resultcol of the same type is named *_1*, the 3rd *_2*, ...
- **timeReq2nd (defining the MX-Resultcolumn-Set):**
 - **TIMESTAMP**
 - if *None* last TIME in *Mx* is used
 - if *aggReq* considered as **TIMESTAMPR** (ignored if *aggReq* = **TIME**)

viewList: Views with MX-Resultcolumn-Set to be added:

- in the Xm-Views below col *mx2Idx* must exist (i.e. *MxSync* mus have been called)
- *mx2Idx* is considered to be the last of the Model-cols
- right from *mx2Idx* all available Result-cols are added if not already existing

- already existing Result-cols are overwritten
- **mx2Idx-Views:**
 - vKNOT (KNOT...)
 - vROHR (ROHR...) - only Non-VEC-Channel-Results are added
 - vFWVB (FWVB...)
 - vVBEL (KNOT..._i and KNOT..._k and Q)
- NEW 1st Call:
 - **vROHRVecResults: VEC-Channel-Results for Pipe-Interior-Pts (IPts):**
 - * pk
 - * mx2Idx
 - * IptIdx: S,0,...,E - Interior Point Index; S=Start EdgeDefNode, E=End EdgeDefNode, 0=1st Ipt in EdgeDefDirection
 - * one column per VEC-Channel
 - **vAGSN**
 - * from vVBEL: KNOT..._i and KNOT..._k and Q
 - * from vROHRVecResults: vecResults
 - * **Topology:**
 - nextNODE
 - IptIdx
 - * **Geometry:**
 - dx
 - x
 - xVbel
 - Z (the corresponding Z_i, Z_k and ZVEC are dropped)
 - * **Results:**
 - Q: from Q before and QMVEC for PIPEs; in Schnittrichtung; QMVEC is dropped
 - **for available KNOT...#_i, KNOT...#_k and ...#VEC:**
 - i.e. KNOT~*~*~*P_i KNOT~*~*~*P_k ROHR~*~*~*PVEC
 - P is new column
 - the correspondig 3 columns are dropped

Returns: Mx-Object if no Mx-Object was given; Nothing else

Raises: XmError

```
>>> # -q -m 0 -t before -s Xm.MxAdd -y yes -z yes -w LocalHeatingNetwork -w_
↳ GPipes
>>> import pandas as pd
>>> pd.set_option('display.max_columns', None)
>>> pd.set_option('display.max_rows', None)
>>> pd.set_option('display.max_colwidth', 666666)
```

(continues on next page)

(continued from previous page)

```

>>> pd.set_option('display.width', 666666666)
>>> xm=xms['LocalHeatingNetwork']
>>> (wDir,modelDir,modelName,mx1File)=xm.getWDirModelDirModelName()
>>> try:
...     import Mx
... except:
...     from PT3S import Mx
>>> mx=Mx.Mx(mx1File=mx1File)
>>> mx.dfVecAggs.loc([( 'TIME', 'TMIN', 'TMAX'], 'KNOT~*~*~*~PH', slice(None),
→ slice(None)), 0:3].round(1).reset_index()

```

	TYPE	Sir3sID	TIMESTAMPL	TIMESTAMPR	0	1	2
→2	3						
0	TIME	KNOT~*~*~*~PH	2004-09-22 08:30:00	2004-09-22 08:30:00	2.3	4.0	4.1
→1	4.1						
1	TMIN	KNOT~*~*~*~PH	2004-09-22 08:30:00	2004-09-22 08:31:00	2.1	2.2	2.3
→2	2.2						
2	TMAX	KNOT~*~*~*~PH	2004-09-22 08:30:00	2004-09-22 08:31:00	2.3	4.0	4.1
→1	4.1						

```

>>> print(xm._getvXXXXAsOneString(vXXXX='vKNOT',filterCollist=['BESCHREIBUNG
→', 'IDREFERENZ', 'NAME', 'KNOT~*~*~*~PH'],roundDct={'KNOT~*~*~*~PH':1}))

```

	BESCHREIBUNG	IDREFERENZ	NAME	KNOT~*~*~*~PH
0	None	-1	R-K004	2.3
1	None	-1	V-K002	4.0
2	None	-1	V-K001	4.1
3	None	-1	V-K000	4.1
4	None	-1	R-K001	2.0
5	None	-1	R-K003	2.3
6	None	-1	R-K000	2.0
7	None	-1	R-K005	2.3
8	None	-1	R-L	2.0
9	None	-1	R-K002	2.1
10	None	-1	V-K004	3.8
11	None	-1	V-K005	3.8
12	None	-1	R-K007	2.3
13	None	-1	V-K006	3.8
14	None	-1	R-K006	2.3
15	None	-1	V-K003	3.8
16	None	-1	V-L	4.1
17	None	-1	V-K007	3.8
18	None	-1	R2	4.3
19	None	-1	V-1	4.1
20	None	-1	R3	4.3
21	Druckhaltung - 2 bar Ruhedruck	-1	PKON-Knoten	2.0
22	Anbindung Druckhaltung	-1	R-1	2.0

```

>>> print(xm._getvXXXXAsOneString(vXXXX='vROHR',filterCollist=['BESCHREIBUNG
→', 'IDREFERENZ', 'NAME_i', 'NAME_k', 'ROHR~*~*~*~QMAV'],roundDct={'ROHR~*~*~*~
→QMAV':1}))

```

	BESCHREIBUNG	IDREFERENZ	NAME_i	NAME_k	ROHR~*~*~*~QMAV
0	None	-1	R-K004	R-K005	-8.5
1	None	-1	V-K002	V-K003	19.1
2	None	-1	R-K003	R-K004	-15.4
3	None	-1	V-K004	V-K005	8.5
4	None	-1	V-K001	V-K002	23.0
5	None	-1	R-K006	R-K007	-3.9
6	None	-1	V-K000	V-K001	23.0
7	None	-1	V-K003	V-K004	15.4
8	None	-1	V-K005	V-K006	3.9

(continues on next page)

(continued from previous page)

```

9      None      -1  R-K001  R-K002      -23.0
10     None      -1  R-K002  R-K003      -19.1
11     None      -1  R-K005  R-K006      -3.9
12     None      -1  V-K006  V-K007       3.9
13     None      -1  R-K000  R-K001      -23.0
14     None      -1      R-L  R-K000      -23.0
15     None      -1      V-L  V-K000      23.0
>>> print(xm._getvXXXXAsOneString(vXXXX='vFWVB'))
    BESCHREIBUNG IDREFERENZ  W0 LFK  WOLFK  TVL0  TRS0  LFKT      W  W_min
→W_max  INDTR  TRSK  VTYP  DPHAUS  IMBG  IRFV      pk
→      tk  NAME_i  KVR_i  TM_i  XKOR_i  YKOR_i  ZKOR_i  pXCor_i  pYCor_i
→NAME_k  KVR_k  TM_k  XKOR_k  YKOR_k  ZKOR_k  pXCor_k  pYCor_k
→      CONT  CONT_ID  CONT_LFDNR
→WBLZ  mx2Idx  FWVB~*~*~*~W  FWVB~*~*~*~QM  FWVB~*~*~*~IAKTIV
0      1      -1  200  0.8  160.0  90  50  LFKT  160.0  160.0
→160.0  1      55  14      0.7  0  0.0  4643800032883366034
→4643800032883366034  V-K002  1  90  2541059  5706265  20  319.0
→ 56.0  R-K002  2  60  2541059  5706265  20  319.0  56.0
→Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1  [BLNZ1,
→BLNZ1u5, BLNZ1u5u7]  0  160.0  3.928166  0.0
1      3      -1  200  1.0  200.0  90  65  LFKT  200.0  200.0
→200.0  1      65  14      0.7  0  0.0  4704603947372595298
→4704603947372595298  V-K004  1  90  2541539  5706361  20  799.0
→ 152.0  R-K004  2  60  2541539  5706361  20  799.0  152.0
→Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1
→      []  1  200.0  6.869426  0.0
2      4      -1  200  0.8  160.0  90  60  LFKT  160.0  160.0
→160.0  1      60  14      0.7  0  0.0  5121101823283893406
→5121101823283893406  V-K005  1  90  2541627  5706363  20  887.0
→ 154.0  R-K005  2  60  2541627  5706363  20  887.0  154.0
→Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1  [BLNZ1u5,
→BLNZ1u5u7, BLNZ5]  2  160.0  4.581308  0.0
3      5      -1  200  0.8  160.0  90  55  LFKT  160.0  160.0
→160.0  1      55  14      0.7  0  0.0  5400405917816384862
→5400405917816384862  V-K007  1  90  2541899  5706325  20  1159.0
→ 116.0  R-K007  2  60  2541899  5706325  20  1159.0  116.0
→Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1
→  [BLNZ1u5u7]  3  160.0  3.928166  0.0
4      2      -1  200  0.6  120.0  90  60  LFKT  120.0  120.0
→120.0  1      62  14      0.7  0  0.0  5695730293103267172
→5695730293103267172  V-K003  1  90  2541457  5706345  20  717.0
→ 136.0  R-K003  2  60  2541457  5706345  20  717.0  136.0
→Nahwärmenetz mit 1000 kW Anschlussleistu  1001  -1
→      []  4  120.0  3.680879  0.0
>>> print(xm._getvXXXXAsOneString(vXXXX='vVBEL',filterColList=['BESCHREIBUNG
→','IDREFERENZ','NAME_i','NAME_k','Q']))
      BESCHREIBUNG IDREFERENZ
→NAME_i  NAME_k      Q
OBJTYPE OBJID
→
FWES  5638756766880678918  BHKW - Modul - 1000 kW therm.  -1
→  R3      V-1      22.9879
FWVB  4643800032883366034      1  -1
→V-K002  R-K002      3.92817
      4704603947372595298      3  -1
→V-K004  R-K004      6.86943
      5121101823283893406      4  -1
→V-K005  R-K005      4.58131

```

(continues on next page)

(continued from previous page)

	5400405917816384862	5	-1	└
→V-K007	R-K007 3.92817			
	5695730293103267172	2	-1	└
→V-K003	R-K003 3.68088			
KLAP	4801110583764519435	None	-1	└
→	R2 R3 22.9879			
PGRP	4986517622672493603	Pumpengruppe	-1	└
→	R-1 R3 None			
PUMP	5481331875203087055	Umwälzpumpe	-1	└
→	R-1 R2 22.9879			
ROHR	4613782368750024999	None	-1	└
→R-K004	R-K005 -8.50947			
	4614949065966596185	None	-1	└
→V-K002	V-K003 19.0598			
	4637102239750163477	None	-1	└
→R-K003	R-K004 -15.3789			
	4713733238627697042	None	-1	└
→V-K004	V-K005 8.50948			
	4769996343148550485	None	-1	└
→	R-L R-K000 -22.9879			
	4789218195240364437	None	-1	└
→V-K001	V-K002 22.9879			
	4939422678063487923	None	-1	└
→	V-L V-K000 22.9879			
	4945727430885351042	None	-1	└
→R-K006	R-K007 -3.92817			
	4984202422877610920	None	-1	└
→V-K000	V-K001 22.9879			
	5037777106796980248	None	-1	└
→V-K003	V-K004 15.3789			
	5123819811204259837	None	-1	└
→V-K005	V-K006 3.92817			
	5266224553324203132	None	-1	└
→R-K001	R-K002 -22.9879			
	5379365049009065623	None	-1	└
→R-K002	R-K003 -19.0598			
	5611703699850694889	None	-1	└
→R-K005	R-K006 -3.92817			
	5620197984230756681	None	-1	└
→V-K006	V-K007 3.92817			
	5647213228462830353	None	-1	└
→R-K000	R-K001 -22.9879			
VENT	4678923650983295610	None	-1	└
→	V-1 V-L 22.9879			
	4897018421024717974	None	-1	└
→	R-L R-1 22.9879			
	5525310316015533093	None	-1	PKON-
→Knoten	R-1 2.19997e-06			
>>> print(xm._getvXXXXAsOneString(vXXXX='vROHRVecResults'				
..., filterColList=['pk', 'mx2Idx', 'IptIdx', 'ROHR~~~~~RHOVEC', 'ROHR~~~~~				
→~TVEC', 'ROHR~~~~~MVEC', 'ROHR~~~~~SVEC', 'ROHR~~~~~PVEC', 'ROHR~~~~~				
→ZVEC']				
..., roundDct={'ROHR~~~~~RHOVEC':2, 'ROHR~~~~~TVEC':1, 'ROHR~~~~~MVEC				
→':1, 'ROHR~~~~~SVEC':2, 'ROHR~~~~~PVEC':1, 'ROHR~~~~~ZVEC':1}				
...))				
	pk mx2Idx IptIdx ROHR~~~~~RHOVEC ROHR~~~~~TVEC			└
→ROHR~~~~~MVEC	ROHR~~~~~SVEC	ROHR~~~~~PVEC	ROHR~~~~~ZVEC	

(continues on next page)

(continued from previous page)

0	4613782368750024999	0	S	983.7	60.0	
→	-2.4	0.00		3.3	20.0	
1	4613782368750024999	0	E	983.7	60.0	
→	-2.4	88.02		3.3	20.0	
2	4614949065966596185	1	S	965.7	90.0	
→	5.3	0.00		5.0	20.0	
3	4614949065966596185	1	E	965.7	90.0	
→	5.3	405.96		4.8	20.0	
4	4637102239750163477	2	S	983.7	60.0	
→	-4.3	0.00		3.3	20.0	
5	4637102239750163477	2	E	983.7	60.0	
→	-4.3	83.55		3.3	20.0	
6	4713733238627697042	3	S	965.7	90.0	
→	2.4	0.00		4.8	20.0	
7	4713733238627697042	3	E	965.7	90.0	
→	2.4	88.02		4.8	20.0	
8	4789218195240364437	5	S	965.7	90.0	
→	6.4	0.00		5.1	20.0	
9	4789218195240364437	5	E	965.7	90.0	
→	6.4	195.53		5.0	20.0	
10	4945727430885351042	7	S	983.7	60.0	
→	-1.1	0.00		3.3	20.0	
11	4945727430885351042	7	E	983.7	60.0	
→	-1.1	109.77		3.3	20.0	
12	4984202422877610920	8	S	965.7	90.0	
→	6.4	0.00		5.1	20.0	
13	4984202422877610920	8	E	965.7	90.0	
→	6.4	76.40		5.1	20.0	
14	5037777106796980248	9	S	965.7	90.0	
→	4.3	0.00		4.8	20.0	
15	5037777106796980248	9	E	965.7	90.0	
→	4.3	83.55		4.8	20.0	
16	5123819811204259837	10	S	965.7	90.0	
→	1.1	0.00		4.8	20.0	
17	5123819811204259837	10	E	965.7	90.0	
→	1.1	164.91		4.8	20.0	
18	5266224553324203132	11	S	983.7	60.0	
→	-6.4	0.00		3.0	20.0	
19	5266224553324203132	11	E	983.7	60.0	
→	-6.4	195.53		3.1	20.0	
20	5379365049009065623	12	S	983.7	60.0	
→	-5.3	0.00		3.1	20.0	
21	5379365049009065623	12	E	983.7	60.0	
→	-5.3	405.96		3.3	20.0	
22	5611703699850694889	13	S	983.7	60.0	
→	-1.1	0.00		3.3	20.0	
23	5611703699850694889	13	E	983.7	60.0	
→	-1.1	164.91		3.3	20.0	
24	5620197984230756681	14	S	965.7	90.0	
→	1.1	0.00		4.8	20.0	
25	5620197984230756681	14	E	965.7	90.0	
→	1.1	109.77		4.8	20.0	
26	5647213228462830353	15	S	983.7	60.0	
→	-6.4	0.00		3.0	20.0	
27	5647213228462830353	15	E	983.7	60.0	
→	-6.4	76.40		3.0	20.0	
28	4769996343148550485	4	S	983.7	60.0	
→	-6.4	0.00		3.0		

(continues on next page)

(continued from previous page)

```

29 4769996343148550485      4      E      983.7      60.0
   ↪      -6.4      73.42      3.0      20.0
30 4939422678063487923      6      S      965.7      90.0
   ↪      6.4      0.00      5.1      20.0
31 4939422678063487923      6      E      965.7      90.0
   ↪      6.4      68.60      5.1      20.0
>>> mx.dfVecAggs.shape # unverändert a
(123, 32)
>>> xm.MxAdd(mx=mx, aggReq='TMAX', ForceNoH5Update=True)
>>> mx.dfVecAggs.shape # unverändert b
(123, 32)
>>> dfTMax=xm.dataFrames['vROHRVecResults'].copy()
>>> xm.MxAdd(mx=mx, aggReq='TMIN', ForceNoH5Update=True)
>>> mx.dfVecAggs.shape # unverändert c
(123, 32)
>>> dfTMin=xm.dataFrames['vROHRVecResults'].copy()
>>> xm.MxAdd(mx=mx, aggReq='MAX', ForceNoH5Update=True) # erzeugt MIN/MAX/DIF
>>> mx.dfVecAggs.shape # doppelt a
(246, 32)
>>> dfMax=xm.dataFrames['vROHRVecResults'].copy()
>>> xm.MxAdd(mx=mx, aggReq='MIN', ForceNoH5Update=True) # ueberfluessig
>>> mx.dfVecAggs.shape # doppelt b
(246, 32)
>>> dfMin=xm.dataFrames['vROHRVecResults'].copy()
>>> import pandas as pd
>>> decimals=pd.Series([6], index=['ROHR~*~*~*~PVEC'])
>>> dfTMax.round(decimals=decimals).equals(dfMax.round(decimals=decimals))
True
>>> dfTMin.round(decimals=decimals).equals(dfMin.round(decimals=decimals))
True
>>> # ----- einzelne Zeiten
>>> r,c=mx.dfVecAggs.shape
>>> xm.MxAdd(mx=mx, aggReq='TIME', ForceNoH5Update=True)
>>> rn,cn=mx.dfVecAggs.shape
>>> (rn,cn)==(r,c)
True
>>> xm.MxAdd(mx=mx, aggReq='TIME', timeReq=mx.df.index[3], ForceNoH5Update=True)
>>> rn,cn=mx.dfVecAggs.shape
>>> (r,c)
(246, 32)
>>> (rn,cn) # 41 neue Einträge = 123/3
(287, 32)
>>> xm.MxAdd(mx=mx, aggReq='TIME', timeReq=mx.df.index[3], ForceNoH5Update=True)
>>> (rn,cn)==mx.dfVecAggs.shape
True
>>> mx=xm.MxAdd(ForceNoH5Update=True)
>>> mx.dfVecAggs.shape # h5-Inhalt unverändert 1
(123, 32)
>>> # --- mehrere Zeiten/Aggs
>>> wDir,modelDir,modelName,mx1Filename = xm.getWDirModelDirModelName()
>>> try:
...     import Mx
... except:
...     from PT3S import Mx
>>> mx = None
>>> mx=Mx.Mx(mx1File=mx1Filename)
>>> mx.dfVecAggs.shape # h5-Inhalt unverändert 2

```

(continues on next page)

(continued from previous page)

```

(123, 32)
>>> xm.MxAdd(mx=mx, aggReq=['TIME', 'TMIN', 'TMAX'], timeReq=[mx.df.index[0], mx.
→df.index[0], mx.df.index[0]], timeReq2nd=[mx.df.index[0], mx.df.index[-1], mx.
→df.index[-1]], ForceNoH5Update=True)
>>> mx.dfVecAggs.shape
(123, 32)
>>> xm.MxAdd(mx=mx, aggReq=['TIME', 'MIN'], timeReq=[mx.df.index[3], mx.df.
→index[0]], timeReq2nd=[mx.df.index[3], mx.df.index[-3]], ForceNoH5Update=True)
>>> mx.dfVecAggs.shape
(287, 32)
>>> xm=xms['LocalHeatingNetwork']
>>> mx=xm.MxAdd(ForceNoH5Update=True)
>>> mx.dfVecAggs.shape
(123, 32)
>>> # mx.dfVecAggs.loc[(slice(None), ['KNOT~*~*~*~*P', 'ROHR~*~*~*~*QMI', 'ROHR~*~
→*~*~*PVECMIN_INST'], slice(None), slice(None)), [0, 1, 2, 31]].round(2)
>>> vAGSN=xm.dataFrames['vAGSN']
>>> vAGSN.shape
(32, 55)
>>> xm.MxAdd(mx=mx, aggReq=['TIME', 'TMIN', 'TMAX'], timeReq=3*[mx.df.index[0]],
→timeReq2nd=3*[mx.df.index[-1]], viewList=['vAGSN'], ForceNoH5Update=True)
>>> mx.dfVecAggs.shape
(123, 32)
>>> vAGSN=xm.dataFrames['vAGSN']
>>> vAGSN.shape
(32, 111)
>>> # vAGSN.filter(regex='([\w ]+)(\d+)$').columns
>>> xm.dataFrames['vAGSNTmp']=vAGSN.round(2)
>>> print(xm._getvXXXXAsOneString(vXXXX='vAGSNTmp', filterColList=['P', 'P_1',
→'P_2'], roundDct={'P':1, 'P_1':1, 'P_2':1}))
      P  P_1  P_2
0    5.1  3.2  5.1
1    5.1  3.2  5.1
2    5.1  3.2  5.1
3    5.1  3.2  5.1
4    5.1  3.2  5.1
5    5.0  3.2  5.0
6    5.0  3.2  5.0
7    4.8  3.2  4.8
8    4.8  3.2  4.8
9    4.8  3.2  4.8
10   4.8  3.2  4.8
11   4.8  3.2  4.8
12   4.8  3.2  4.8
13   4.8  3.2  4.8
14   4.8  3.2  4.8
15   4.8  3.2  4.8
16   3.0  3.0  3.0
17   3.0  3.0  3.0
18   3.0  3.0  3.0
19   3.0  3.0  3.0
20   3.0  3.0  3.0
21   3.1  3.0  3.1
22   3.1  3.0  3.1
23   3.3  3.0  3.3
24   3.3  3.0  3.3
25   3.3  3.0  3.3

```

(continues on next page)

(continued from previous page)

```

26 3.3 3.0 3.3
27 3.3 3.0 3.3
28 3.3 3.0 3.3
29 3.3 3.0 3.3
30 3.3 3.0 3.3
31 3.3 3.0 3.3
>>> mx.dfVecAggs.shape
(123, 32)
>>> xm.MxAdd(mx=mx, aggReq=['MIN'], timeReq=1*[mx.df.index[1]],
↳timeReq2nd=1*[mx.df.index[-2]], viewList=['vAGSN'], ForceNoH5Update=True)
>>> mx.dfVecAggs.shape
(246, 32)
>>> xm.MxAdd(mx=mx, aggReq=['MAX'], timeReq=1*[mx.df.index[0]],
↳timeReq2nd=1*[mx.df.index[1]], viewList=['vAGSN', 'vKNOT', 'vFWVB', 'vROHR'],
↳ForceNoH5Update=True)
>>> mx.dfVecAggs.shape
(369, 32)
>>> #mx.dfVecAggs.loc[(slice(None), ['KNOT~*~*~*~P', 'ROHR~*~*~*~QMI', 'ROHR~*~
↳*~*~PVECMIN_INST'], slice(None), slice(None)), [0, 1, 2, 31]].round(2)
>>> xm.MxAdd(mx=mx
... , aggReq=['TIME']
... , timeReq=1*[mx.df.index[0]]
... , timeReq2nd=[None]
... , viewList=['vAGSN']
... , ForceNoH5Update=True)
>>> xm.MxAdd(mx=mx
... , aggReq=['TIME']
... , timeReq=1*[mx.df.index[0]]
... , timeReq2nd=[None]
... , viewList=['vKNOT']
... , ForceNoH5Update=True)
>>> xm.MxAdd(mx=mx
... , aggReq=['TIME']
... , timeReq=1*[mx.df.index[0]]
... , timeReq2nd=[None]
... , viewList=['vROHR']
... , ForceNoH5Update=True)
>>> xm.MxAdd(mx=mx
... , aggReq=['TIME']
... , timeReq=1*[mx.df.index[0]]
... , timeReq2nd=[None]
... , viewList=['vFWVB']
... , ForceNoH5Update=True)
>>> xm.MxAdd(mx=mx
... , aggReq=['TIME']
... , timeReq=1*[mx.df.index[0]]
... , timeReq2nd=[None]
... , viewList=['vVBEL']
... , ForceNoH5Update=True)
>>> ###
>>> aggReqLst=['TIME', 'TMIN', 'TMAX', 'TIME']
>>> timeReqLst=3*[mx.df.index[0]]+[mx.df.index[-1]]
>>> timeReq2ndLst=4*[mx.df.index[-1]]
>>> viewLst=['vAGSN', 'vKNOT', 'vROHR', 'vFWVB', 'vVBEL']
>>> xm.MxAdd(mx=mx
... , aggReq=aggReqLst
... , timeReq=timeReqLst

```

(continues on next page)

(continued from previous page)

```

... ,timeReq2nd=timeReq2ndLst
... ,viewList=viewLst
... ,ForceNoH5Update=True)
>>> vKNOT=xm.dataFrames['vKNOT']
>>> xm.dataFrames['vKNOTTmp']=vKNOT.round(2)
>>> print(xm._getvXXXXAsOneString(vXXXX='vKNOTTmp',filterColList=['KNOT~*~*~
→~*~PH','KNOT~*~*~*~PH_1','KNOT~*~*~*~PH_2','KNOT~*~*~*~PH_3'],roundDct={
→'KNOT~*~*~*~PH':1,'KNOT~*~*~*~PH_1':1,'KNOT~*~*~*~PH_2':1,'KNOT~*~*~*~PH_3
→':1}))

```

	KNOT~*~*~*~PH	KNOT~*~*~*~PH_1	KNOT~*~*~*~PH_2	KNOT~*~*~*~PH_3
0	2.3	2.0	2.3	2.3
1	4.0	2.2	4.0	4.0
2	4.1	2.2	4.1	4.1
3	4.1	2.2	4.1	4.1
4	2.0	2.0	2.0	2.0
5	2.3	2.0	2.3	2.3
6	2.0	2.0	2.0	2.0
7	2.3	2.0	2.3	2.3
8	2.0	2.0	2.0	2.0
9	2.1	2.0	2.1	2.1
10	3.8	2.2	3.8	3.8
11	3.8	2.2	3.8	3.8
12	2.3	2.0	2.3	2.3
13	3.8	2.2	3.8	3.8
14	2.3	2.0	2.3	2.3
15	3.8	2.2	3.8	3.8
16	4.1	2.2	4.1	4.1
17	3.8	2.2	3.8	3.8
18	4.3	2.2	4.3	4.3
19	4.1	2.2	4.1	4.1
20	4.3	2.2	4.3	4.3
21	2.0	2.0	2.0	2.0
22	2.0	2.0	2.0	2.0

```

>>> xm.MxAdd(mx=mx)
>>> vKNOT=xm.dataFrames['vKNOT']
>>> xm.dataFrames['vKNOTTmp']=vKNOT.round(2)
>>> print(xm._getvXXXXAsOneString(vXXXX='vKNOTTmp',filterColList=['KNOT~*~*~
→~*~PH','KNOT~*~*~*~PH_1','KNOT~*~*~*~PH_2','KNOT~*~*~*~PH_3']))

```

	KNOT~*~*~*~PH
0	2.30
1	3.99
2	4.08
3	4.12
4	2.04
5	2.28
6	2.00
7	2.31
8	2.00
9	2.14
10	3.83
11	3.82
12	2.31
13	3.82
14	2.31
15	3.85
16	4.13
17	3.81

(continues on next page)

(continued from previous page)

18	4.31
19	4.13
20	4.29
21	2.00
22	2.00

_MxAddvVBEL (*dfSource*)

(Re-)constructing vVBEL with MX2-Results.

Arguments: *dfSource***Result:****View with MX2-Results added:**

- in the Xm-View below col `mx2Idx` must exist
- `mx2Idx` is considered to be the last of the Model-cols
- right from `mx2Idx` Result-cols are added if not already existing
- already existing Result-cols are overwritten
- **mx2Idx-View:**
 - ...
 - vVBEL (KNOT...`_i` and KNOT...`_k` and Q)

Raises: `XmError`

```

>>> xm=xms ['GPipes']
>>> mx=xm.MxSync ()
>>> xm.MxAdd (mx=mx)
>>> print (xm._getvXXXXAsOneString (vXXXX='vVBEL',filterColList=['BESCHREIBUNG
→', 'IDREFERENZ', 'tk NAME_i', 'CONT_i', ' CONT_VKNO_i', 'Z_i', 'pk_i']))
                                BESCHREIBUNG          IDREFERENZ    CONT_i    Z_
→i                                pk_i
OBJTYPE OBJID
→
ROHR      4979507900871287244  _Split_Split  3S4979507900871287244  M-1-0-1
→0  4731210032713520411
      5114681686941855110          _Split  3S5114681686941855110  M-1-0-1
→0  5709889458254995435
      5244313507655010738          NaN  3S5244313507655010738  M-1-0-1
→0  5256558483525770176
      5694016449043789006          NaN  3S5694016449043789006  M-1-0-1
→0  4731210032713520411
VENT      5116489323526156845          NaN  3S5508684139418025293  M-1-0-1
→100  4683988347517083361
      5309992331398639768          NaN  3S5309992331398639768  M-1-0-1
→0  5046108271210239718
      5508684139418025293          NaN  3S5508684139418025293  M-1-0-1
→100  4683988347517083361
      5745097345184516675          NaN  3S5745097345184516675  M-1-0-1
→0  5308591811899364960
>>> print (xm._getvXXXXAsOneString (vXXXX='vVBEL',filterColList=['BESCHREIBUNG
→', 'IDREFERENZ', 'tk NAME_k', 'CONT_k', ' CONT_VKNO_k', 'Z_k', 'pk_k']))
                                BESCHREIBUNG          IDREFERENZ    CONT_k    Z_
→k                                pk_k

```

(continues on next page)

(continued from previous page)

```

OBJTYPE OBJID
↳
ROHR      4979507900871287244  _Split_Split  3S4979507900871287244  M-1-0-1
↳0  5441322867018839631
      5114681686941855110      _Split  3S5114681686941855110  M-1-0-1
↳0  5441322867018839631
      5244313507655010738      NaN  3S5244313507655010738  M-1-0-1
↳100  4683988347517083361
      5694016449043789006      NaN  3S5694016449043789006  M-1-0-1
↳0  5061043246189134395
VENT      5116489323526156845      NaN  3S5508684139418025293  M-1-0-1
↳0  5709889458254995435
      5309992331398639768      NaN  3S5309992331398639768  M-1-0-1
↳0  5256558483525770176
      5508684139418025293      NaN  3S5508684139418025293  M-1-0-1
↳0  5709889458254995435
      5745097345184516675      NaN  3S5745097345184516675  M-1-0-1
↳0  4731210032713520411
>>> print(xm._getvXXXXAsOneString(vXXXX='vVBEL',filterColList=['LAYR','L','D
↳','mx2Idx']))

```

		LAYR	L	D	mx2Idx
OBJTYPE	OBJID				
ROHR	4979507900871287244	[]	10000	500	2
	5114681686941855110	[]	10000	500	1
	5244313507655010738	[]	160000	500	0
	5694016449043789006	[]	100.498688	450	3
VENT	5116489323526156845	[]	0	666	3
	5309992331398639768	[]	0	800	0
	5508684139418025293	[]	0	800	2
	5745097345184516675	[]	0	800	1

```

>>> print(xm._getvXXXXAsOneString(vXXXX='vVBEL',filterColList=['KNOT~*****
↳RHON_i','KNOT~*****H_i','KNOT~*****LFAKTAKT_i','KNOT~*****P_i','KNOT~*
↳*****PH_i','KNOT~*****PH_EIN_i','KNOT~*****QM_i','KNOT~*****RHO_i','KNOT~
↳*****T_i','KNOT~*****EH_i'])
...      ,roundDct={'KNOT~*****H_i':1,'KNOT~*****P_i':1,'KNOT~*****PH_i
↳':1,'KNOT~*****PH_EIN_i':1,'KNOT~*****QM_i':1,'KNOT~*****RHO_i':1,'KNOT~
↳*****T_i':1,'KNOT~*****EH_i':1})
...      )

```

		KNOT~*****RHON_i	KNOT~*****H_i	KNOT~*****LFAKTAKT_i	KNOT~*****P_i	KNOT~*****PH_i	KNOT~*****PH_EIN_i	KNOT~*****QM_i	KNOT~*****RHO_i	KNOT~*****T_i	KNOT~*****EH_i
OBJTYPE	OBJID										
ROHR	4979507900871287244		0.83			10106.2					
↳	1.0		11.1		10.1		10.1				
↳ 0.0			7.9		49.0		13046.7				
	5114681686941855110		0.83			16375.8					
↳	1.0		17.4		16.4		16.4				
↳ 0.0			12.8		40.9		13008.5				
	5244313507655010738		0.83			39974.0					
↳	1.0		41.0		40.0		40.0				
↳ 0.0			31.9		40.0		12777.4				
	5694016449043789006		0.83			10106.2					
↳	1.0		11.1		10.1		10.1				
↳ 0.0			7.9		49.0		13046.7				

(continues on next page)

(continued from previous page)

```

VENT      5116489323526156845      0.83      16440.5
→      1.0      17.4      16.4      16.4
→ 0.0      12.9      40.9      13111.2
      5309992331398639768      0.83      40000.0
→      1.0      41.0      40.0      40.0
→118257.5      31.9      40.0      12777.4
      5508684139418025293      0.83      16440.5
→      1.0      17.4      16.4      16.4
→ 0.0      12.9      40.9      13111.2
      5745097345184516675      0.83      10000.0
→      1.0      11.0      10.0      10.0
→118257.5      7.8      49.0      13034.1
>>> print(xm._getvXXXXAsOneString(vXXXX='vVBEL',filterColList=['KNOT~*~*~*~*~
→HMAX_INST_i','KNOT~*~*~*~*~HMIN_INST_i','KNOT~*~*~*~*~PMAX_INST_i','KNOT~*~*~*~*~
→PMIN_INST_i','KNOT~*~*~*~*~IAKTIV_i','KNOT~*~*~*~*~PDAMPF_i']
...      ,roundDct={'KNOT~*~*~*~*~HMAX_INST_i':1,'KNOT~*~*~*~*~HMIN_INST_i':1,
→'KNOT~*~*~*~*~PMAX_INST_i':1,'KNOT~*~*~*~*~PMIN_INST_i':1})
...      ))
      KNOT~*~*~*~*~HMAX_INST_i  KNOT~*~*~*~*~HMIN_INST_i
→KNOT~*~*~*~*~PMAX_INST_i  KNOT~*~*~*~*~PMIN_INST_i  KNOT~*~*~*~*~IAKTIV_i  KNOT~
→*~*~*~*~PDAMPF_i
OBJTYPE OBJID
→
→
ROHR      4979507900871287244      10106.2      10106.2
→      11.1      11.1      0.0
→      0.0
      5114681686941855110      16375.8      16375.8
→      17.4      17.4      0.0
→      0.0
      5244313507655010738      39974.0      39974.0
→      41.0      41.0      0.0
→      0.0
      5694016449043789006      10106.2      10106.2
→      11.1      11.1      0.0
→      0.0
VENT      5116489323526156845      16440.5      16440.5
→      17.4      17.4      0.0
→      0.0
      5309992331398639768      40000.0      40000.0
→      41.0      41.0      0.0
→      0.0
      5508684139418025293      16440.5      16440.5
→      17.4      17.4      0.0
→      0.0
      5745097345184516675      10000.0      10000.0
→      11.0      11.0      0.0
→      0.0

```

_MxAddvROHRVecResults (*dfSource*)
(Re-)constructing vROHRVecResults.

Arguments:

- dfSource

Result:

- **vROHRVecResults: VEC-Channel-Results for Pipe-Interior-Pts (IPts):**

- Raises:**
- XmError

(continues on next page)

(continued from previous page)

17	5244313507655010738	0	8	45000.0	40.3	
→	28.1	36.0		27.3	27.7	
→	35.0	3540.0				
18	5244313507655010738	0	9	50000.0	40.3	
→	31.2	35.4		27.3	27.2	
→	34.4	3604.2				
19	5244313507655010738	0	10	55000.0	40.3	
→	34.4	34.8		27.3	26.7	
→	33.8	3671.8				
20	5244313507655010738	0	11	60000.0	40.3	
→	37.5	34.2		27.3	26.2	
→	33.2	3743.2				
21	5244313507655010738	0	12	65000.0	40.3	
→	40.6	33.5		27.3	25.7	
→	32.5	3818.6				
22	5244313507655010738	0	13	70000.0	40.3	
→	43.8	32.9		27.3	25.2	
→	31.9	3898.6				
23	5244313507655010738	0	14	75000.0	40.3	
→	46.9	32.2		27.3	24.6	
→	31.2	3983.5				
24	5244313507655010738	0	15	80000.0	40.3	
→	50.0	31.6		27.3	24.1	
→	30.6	4073.9				
25	5244313507655010738	0	16	85000.0	40.3	
→	53.1	30.9		27.3	23.5	
→	29.9	4170.5				
26	5244313507655010738	0	17	90000.0	40.3	
→	56.2	30.2		27.3	23.0	
→	29.2	4273.9				
27	5244313507655010738	0	18	95000.0	40.3	
→	59.4	29.5		27.3	22.4	
→	28.5	4385.1				
28	5244313507655010738	0	19	100000.0	40.3	
→	62.5	28.7		27.3	21.8	
→	27.7	4505.0				
29	5244313507655010738	0	20	105000.0	40.3	
→	65.6	28.0		27.3	21.2	
→	27.0	4634.8				
30	5244313507655010738	0	21	110000.0	40.3	
→	68.8	27.2		27.3	20.6	
→	26.2	4776.0				
31	5244313507655010738	0	22	115000.0	40.4	
→	71.9	26.4		27.3	19.9	
→	25.4	4930.3				
32	5244313507655010738	0	23	120000.0	40.4	
→	75.0	25.5		27.3	19.2	
→	24.5	5100.1				
33	5244313507655010738	0	24	125000.0	40.4	
→	78.1	24.7		27.3	18.6	
→	23.7	5287.9				
34	5244313507655010738	0	25	130000.0	40.4	
→	81.2	23.8		27.3	17.9	
→	22.8	5497.2				
35	5244313507655010738	0	26	135000.0	40.5	
→	84.4	22.9		27.3	17.1	
→	21.9	5732.6				

(continues on next page)

(continued from previous page)

```

36 5244313507655010738      0      27      140000.0      40.5
→      87.5      21.9      27.3      16.4
→      20.9      5999.9
37 5244313507655010738      0      28      145000.0      40.6
→      90.6      20.9      27.3      15.6
→      19.9      6307.0
38 5244313507655010738      0      29      150000.0      40.7
→      93.8      19.8      27.3      14.7
→      18.8      6665.0
39 5244313507655010738      0      30      155000.0      40.8
→      96.9      18.7      27.3      13.8
→      17.7      7089.7
40 5244313507655010738      0      E      160000.0      40.9
→      100.0      17.4      27.3      12.9
→      16.4      7604.8
5 5114681686941855110      1      S      0.0      40.9
→      0.0      17.4      27.3      12.8
→      16.4      7652.5
6 5114681686941855110      1      0      5000.0      43.7
→      0.0      16.1      27.3      11.8
→      15.1      8353.3
4 4979507900871287244      2      E      10000.0      44.0
→      0.0      14.6      -27.3      10.6
→      13.6      -9242.2
7 5114681686941855110      1      E      10000.0      44.0
→      0.0      14.6      27.3      10.7
→      13.6      9214.7
3 4979507900871287244      2      0      5000.0      48.2
→      0.0      13.0      -27.3      9.3
→      12.0      -10534.3
0 5694016449043789006      3      S      0.0      10.0
→      0.0      11.1      0.0      6.9
→      10.1      0.0
2 4979507900871287244      2      S      0.0      49.0
→      0.0      11.1      -27.3      7.9
→      10.1      -12394.5
1 5694016449043789006      3      E      100.5      10.0
→      0.0      1.0      0.0      0.6
→      0.0      0.0
>>> print(xm._getvXXXXAsOneString(vXXXX='vROHRVecResults',roundDct={'ROHR~*~
→*~*~SVEC':1,'ROHR~*~*~*~TVEC':1,'ROHR~*~*~*~ZVEC':1,'ROHR~*~*~*~PVEC':1,
→'ROHR~*~*~*~MVEC':1,'ROHR~*~*~*~RHOVEC':1,'ROHR~*~*~*~PHVEC':1,'ROHR~*~*~*~
→QMVEC':1}))
      pk  mx2Idx  IptIdx  ROHR~*~*~*~SVEC  ROHR~*~*~*~TVEC
→ROHR~*~*~*~ZVEC  ROHR~*~*~*~PVEC  ROHR~*~*~*~MVEC  ROHR~*~*~*~RHOVEC  ROHR~
→*~*~*~PHVEC  ROHR~*~*~*~QMVEC
0 5694016449043789006      3      S      0.0      10.0
→      0.0      11.1      0.0      6.9
→      10.1      0.0
1 5694016449043789006      3      E      100.5      10.0
→      0.0      1.0      0.0      0.6
→      0.0      0.0
2 4979507900871287244      2      S      0.0      49.0
→      0.0      11.1      -27.3      7.9
→      10.1      -12394.5
3 4979507900871287244      2      0      5000.0      48.2
→      0.0      13.0      -27.3      9.3
→      12.0      -10534.3

```

(continues on next page)

(continued from previous page)

4	4979507900871287244	2	E	10000.0	44.0	
→	0.0	14.6		-27.3	10.6	
→	13.6	-9242.2				
5	5114681686941855110	1	S	0.0	40.9	
→	0.0	17.4		27.3	12.8	
→	16.4	7652.5				
6	5114681686941855110	1	0	5000.0	43.7	
→	0.0	16.1		27.3	11.8	
→	15.1	8353.3				
7	5114681686941855110	1	E	10000.0	44.0	
→	0.0	14.6		27.3	10.7	
→	13.6	9214.7				
8	5244313507655010738	0	S	0.0	40.0	
→	0.0	41.0		27.3	31.9	
→	40.0	3077.8				
9	5244313507655010738	0	0	5000.0	40.4	
→	3.1	40.5		27.3	31.4	
→	39.5	3121.6				
10	5244313507655010738	0	1	10000.0	40.4	
→	6.2	39.9		27.3	31.0	
→	38.9	3166.3				
11	5244313507655010738	0	2	15000.0	40.4	
→	9.4	39.4		27.3	30.5	
→	38.4	3212.9				
12	5244313507655010738	0	3	20000.0	40.4	
→	12.5	38.8		27.3	30.1	
→	37.8	3261.5				
13	5244313507655010738	0	4	25000.0	40.4	
→	15.6	38.3		27.3	29.6	
→	37.3	3312.3				
14	5244313507655010738	0	5	30000.0	40.4	
→	18.8	37.7		27.3	29.2	
→	36.7	3365.3				
15	5244313507655010738	0	6	35000.0	40.4	
→	21.9	37.2		27.3	28.7	
→	36.2	3420.8				
16	5244313507655010738	0	7	40000.0	40.3	
→	25.0	36.6		27.3	28.2	
→	35.6	3478.9				
17	5244313507655010738	0	8	45000.0	40.3	
→	28.1	36.0		27.3	27.7	
→	35.0	3540.0				
18	5244313507655010738	0	9	50000.0	40.3	
→	31.2	35.4		27.3	27.2	
→	34.4	3604.2				
19	5244313507655010738	0	10	55000.0	40.3	
→	34.4	34.8		27.3	26.7	
→	33.8	3671.8				
20	5244313507655010738	0	11	60000.0	40.3	
→	37.5	34.2		27.3	26.2	
→	33.2	3743.2				
21	5244313507655010738	0	12	65000.0	40.3	
→	40.6	33.5		27.3	25.7	
→	32.5	3818.6				
22	5244313507655010738	0	13	70000.0	40.3	
→	43.8	32.9		27.3	25.2	
→	31.9	3898.6				

(continues on next page)

(continued from previous page)

23	5244313507655010738	0	14	75000.0	40.3	┌
→	46.9	32.2		27.3	24.6	┌
→	31.2	3983.5				┌
24	5244313507655010738	0	15	80000.0	40.3	┌
→	50.0	31.6		27.3	24.1	┌
→	30.6	4073.9				┌
25	5244313507655010738	0	16	85000.0	40.3	┌
→	53.1	30.9		27.3	23.5	┌
→	29.9	4170.5				┌
26	5244313507655010738	0	17	90000.0	40.3	┌
→	56.2	30.2		27.3	23.0	┌
→	29.2	4273.9				┌
27	5244313507655010738	0	18	95000.0	40.3	┌
→	59.4	29.5		27.3	22.4	┌
→	28.5	4385.1				┌
28	5244313507655010738	0	19	100000.0	40.3	┌
→	62.5	28.7		27.3	21.8	┌
→	27.7	4505.0				┌
29	5244313507655010738	0	20	105000.0	40.3	┌
→	65.6	28.0		27.3	21.2	┌
→	27.0	4634.8				┌
30	5244313507655010738	0	21	110000.0	40.3	┌
→	68.8	27.2		27.3	20.6	┌
→	26.2	4776.0				┌
31	5244313507655010738	0	22	115000.0	40.4	┌
→	71.9	26.4		27.3	19.9	┌
→	25.4	4930.3				┌
32	5244313507655010738	0	23	120000.0	40.4	┌
→	75.0	25.5		27.3	19.2	┌
→	24.5	5100.1				┌
33	5244313507655010738	0	24	125000.0	40.4	┌
→	78.1	24.7		27.3	18.6	┌
→	23.7	5287.9				┌
34	5244313507655010738	0	25	130000.0	40.4	┌
→	81.2	23.8		27.3	17.9	┌
→	22.8	5497.2				┌
35	5244313507655010738	0	26	135000.0	40.5	┌
→	84.4	22.9		27.3	17.1	┌
→	21.9	5732.6				┌
36	5244313507655010738	0	27	140000.0	40.5	┌
→	87.5	21.9		27.3	16.4	┌
→	20.9	5999.9				┌
37	5244313507655010738	0	28	145000.0	40.6	┌
→	90.6	20.9		27.3	15.6	┌
→	19.9	6307.0				┌
38	5244313507655010738	0	29	150000.0	40.7	┌
→	93.8	19.8		27.3	14.7	┌
→	18.8	6665.0				┌
39	5244313507655010738	0	30	155000.0	40.8	┌
→	96.9	18.7		27.3	13.8	┌
→	17.7	7089.7				┌
40	5244313507655010738	0	E	160000.0	40.9	┌
→	100.0	17.4		27.3	12.9	┌
→	16.4	7604.8				┌
>>> xm=xms ['LocalHeatingNetwork']						
>>> mx=xm.MxSync ()						
>>> xm.MxAdd (mx=mx)						

(continues on next page)

(continued from previous page)

```
>>> print(xm._getvXXXXAsOneString(vXXXX='vROHRVecResults',filterColList=[
→ 'mx2Idx', 'IptIdx', 'ROHR~*~*~*~SVEC'],index=True))
    mx2Idx  IptIdx  ROHR~*~*~*~SVEC
0         0      S      0.000000
1         0      E     88.019997
2         1      S      0.000000
3         1      E    405.959991
4         2      S      0.000000
5         2      E     83.550003
6         3      S      0.000000
7         3      E     88.019997
8         5      S      0.000000
9         5      E    195.529999
10        7      S      0.000000
11        7      E    109.769997
12        8      S      0.000000
13        8      E     76.400002
14        9      S      0.000000
15        9      E     83.550003
16       10      S      0.000000
17       10      E    164.910004
18       11      S      0.000000
19       11      E    195.529999
20       12      S      0.000000
21       12      E    405.959991
22       13      S      0.000000
23       13      E    164.910004
24       14      S      0.000000
25       14      E    109.769997
26       15      S      0.000000
27       15      E     76.400002
28        4      S      0.000000
29        4      E     73.419998
30        6      S      0.000000
31        6      E     68.599998
```

_MxAddvAGSN()

(Re-)constructing vAGSN.

Result:

• **vAGSN**

- all cols from vAGSN_raw
- **from vVBEL:**
 - * KNOT..._i (dropped for processed RVECS)
 - * and KNOT..._k (dropped for processed RVECS)
 - * and Q; Q also contains vROHRVecResults - RVEC QMVEC is dropped; Q in Schnitttrichtung
- **due to vROHRVecResults:**
 - * IptIdx
 - * RVECS not processed, i.e. ROHR~*~*~*MVEC
 - * **Geometry:**

- dx
- x
- xVbel

*** RVECs processed, i.e.**

- RHO - the corresponding KNOT~*~*~RHO_i KNOT~*~*~RHO_k ROHR~*~*~RHOVEC are dropped
- T - the corresponding KNOT~*~*~T_i KNOT~*~*~T_k ROHR~*~*~TVEC are dropped
- P - the corresponding KNOT~*~*~P_i KNOT~*~*~P_k ROHR~*~*~PVEC are dropped

P is new column

the corresponding 3 source-columns are dropped

- and Z (the corresponding Z_i, Z_k and ZVEC are dropped)

Raises: XmError

```
>>> xm=xms['GPipes']
>>> mx=xm.MxSync()
>>> xm.MxAdd(mx=mx)
>>> vAGSN=xm.dataFrames['vAGSN']
>>> schnitt=vAGSN[vAGSN['NAME']=='LR']
>>> schnitt=schnitt.copy()
>>> schnitt.loc[:, 'PH'] = schnitt['PH'].astype(float).values
>>> schnitt.loc[:, 'Q'] = schnitt['Q'].astype(float).values
>>> xm.dataFrames['schnitt']=schnitt
>>> print(xm._getvXXXXAsOneString(vXXXX='schnitt', filterColList=['OBJTYPE',
→ 'NAME_i', 'NAME_k', 'IptIdx', 'nextNODE', 'x', 'PH', 'Q'], roundDct={'PH':1, 'Q':1}
→ , index=True))
```

	OBJTYPE	NAME_i	NAME_k	IptIdx	nextNODE	x	PH	Q
79	VENT	GL	G1	S	G1	0.0	40.0	118257.5
80	VENT	GL	G1	E	G1	0.0	40.0	118257.5
81	ROHR	G1	GKS	S	GKS	0.0	40.0	3077.8
82	ROHR	G1	GKS	0	GKS	5000.0	39.5	3121.6
83	ROHR	G1	GKS	1	GKS	10000.0	38.9	3166.3
84	ROHR	G1	GKS	2	GKS	15000.0	38.4	3212.9
85	ROHR	G1	GKS	3	GKS	20000.0	37.8	3261.5
86	ROHR	G1	GKS	4	GKS	25000.0	37.3	3312.3
87	ROHR	G1	GKS	5	GKS	30000.0	36.7	3365.3
88	ROHR	G1	GKS	6	GKS	35000.0	36.2	3420.8
89	ROHR	G1	GKS	7	GKS	40000.0	35.6	3478.9
90	ROHR	G1	GKS	8	GKS	45000.0	35.0	3540.0
91	ROHR	G1	GKS	9	GKS	50000.0	34.4	3604.2
92	ROHR	G1	GKS	10	GKS	55000.0	33.8	3671.8
93	ROHR	G1	GKS	11	GKS	60000.0	33.2	3743.2
94	ROHR	G1	GKS	12	GKS	65000.0	32.5	3818.6
95	ROHR	G1	GKS	13	GKS	70000.0	31.9	3898.6
96	ROHR	G1	GKS	14	GKS	75000.0	31.2	3983.5
97	ROHR	G1	GKS	15	GKS	80000.0	30.6	4073.9
98	ROHR	G1	GKS	16	GKS	85000.0	29.9	4170.5
99	ROHR	G1	GKS	17	GKS	90000.0	29.2	4273.9
100	ROHR	G1	GKS	18	GKS	95000.0	28.5	4385.1
101	ROHR	G1	GKS	19	GKS	100000.0	27.7	4505.0

(continues on next page)

(continued from previous page)

102	ROHR	G1	GKS	20	GKS	105000.0	27.0	4634.8
103	ROHR	G1	GKS	21	GKS	110000.0	26.2	4776.0
104	ROHR	G1	GKS	22	GKS	115000.0	25.4	4930.3
105	ROHR	G1	GKS	23	GKS	120000.0	24.5	5100.1
106	ROHR	G1	GKS	24	GKS	125000.0	23.7	5287.9
107	ROHR	G1	GKS	25	GKS	130000.0	22.8	5497.2
108	ROHR	G1	GKS	26	GKS	135000.0	21.9	5732.6
109	ROHR	G1	GKS	27	GKS	140000.0	20.9	5999.9
110	ROHR	G1	GKS	28	GKS	145000.0	19.9	6307.0
111	ROHR	G1	GKS	29	GKS	150000.0	18.8	6665.0
112	ROHR	G1	GKS	30	GKS	155000.0	17.7	7089.7
113	ROHR	G1	GKS	E	GKS	160000.0	16.4	7604.8
114	VENT	GKS	GKD	S	GKD	160000.0	16.4	118257.5
115	VENT	GKS	GKD	E	GKD	160000.0	16.4	118257.5
116	ROHR	GKD	G3	S	G3	160000.0	16.4	7652.5
117	ROHR	GKD	G3	0	G3	165000.0	15.1	8353.3
118	ROHR	GKD	G3	E	G3	170000.0	13.6	9214.7
119	ROHR	G4	G3	S	G4	170000.0	13.6	9242.2
120	ROHR	G4	G3	0	G4	175000.0	12.0	10534.3
121	ROHR	G4	G3	E	G4	180000.0	10.1	12394.5
122	VENT	GR	G4	S	GR	180000.0	10.1	118257.5
123	VENT	GR	G4	E	GR	180000.0	10.0	118257.5

__MxAddForOneDf (*dfTarget=None, dfSource=None, multiIndexKey=None, testStr='testStr'*)

Add MX2-Resultdata from dfSource as cols to returned dfTarget.

Args: dfTarget: df with col mx2Idx dfSource: df with mx2Idx-corresponding index and cols (containing MX2-Resultdata) to be added multiIndexKey: value for 1st Index if dfTarget is Multiindexed - i.e. 'XXXX'

Notes:

- all cols from dfSource are added at the end of dfTarget in dfSource-sequence
- the cols can already exist in dfTarget
- if so, _all cols must already exist ...
- ... the dfTarget-sequence should but must be not necessary the dfSource-sequence

Returns: dfTarget

Raises: XmError


```
>>> # ---
>>> # SETUP
>>> # ---
>>> import os
>>> import logging
>>> logger = logging.getLogger('PT3S.Rm')
>>> # ---
>>> # path
>>> # ---
>>> if __name__ == "__main__":
...     try:
...         dummy=__file__
...         logger.debug("{0:s}{1:s}{2:s}".format('DOCTEST: __main__ Context: ', 'path = ',
↪os.path.dirname(__file__)', " ."))
...         path = os.path.dirname(__file__)
...     except NameError:
...         logger.debug("{0:s}{1:s}{2:s}".format('DOCTEST: __main__ Context: ', "path =
↪'.' because __file__ not defined and: ", " from Rm import Rm"))
...         path = '.'
...         from Rm import Rm
...     else:
...         path = '.'
...         logger.debug("{0:s}{1:s}".format('Not __main__ Context: ', "path = '.' ."))
>>> try:
...     from PT3S import Mx
... except ImportError:
...     logger.debug("{0:s}{1:s}".format("DOCTEST: from PT3S import Mx: ImportError: ",
↪"trying import Mx instead ... maybe pip install -e . is active ..."))
...     import Mx
>>> try:
...     from PT3S import Xm
... except ImportError:
...     logger.debug("{0:s}{1:s}".format("DOCTEST: from PT3S import Xm: ImportError: ",
↪"trying import Xm instead ... maybe pip install -e . is active ..."))
...     import Xm
>>> # ---
>>> # testDir
>>> # ---
>>> # globs={'testDir':'testdata'}
>>> try:
...     dummy= testDir
... except NameError:
...     testDir='testdata'
>>> # ---
```

(continues on next page)

(continued from previous page)

```

>>> # dotResolution
>>> # ---
>>> # globs={'dotResolution':''}
>>> try:
...     dummy= dotResolution
... except NameError:
...     dotResolution=''
>>> import pandas as pd
>>> import matplotlib.pyplot as plt
>>> pd.set_option('display.max_columns',None)
>>> pd.set_option('display.width',666666666)
>>> # ---
>>> # LocalHeatingNetwork SETUP
>>> # ---
>>> xmlFile=os.path.join(os.path.join(path,testDir),'LocalHeatingNetwork.XML')
>>> xm=Xm.Xm(xmlFile=xmlFile)
>>> mx1File=os.path.join(path,os.path.join(testDir,'WDLocalHeatingNetwork\B1\V0\BZ1\M-
↳1-0-1'+dotResolution+'.MX1'))
>>> mx=Mx.Mx(mx1File=mx1File,NoH5Read=True,NoMxsRead=True)
>>> mx.setResultsToMxsFile(NewH5Vec=True)
5
>>> xm.MxSync(mx=mx)
>>> rm=Rm(xm=xm,mx=mx)
>>> # ---
>>> # Plot 3Classes False
>>> # ---
>>> plt.close('all')
>>> ppi=72 # matplotlib default
>>> dpi_screen=2*ppi
>>> fig=plt.figure(dpi=dpi_screen,linewidth=1.)
>>> timeDeltaToT=mx.df.index[2]-mx.df.index[0]
>>> # 3Classes und FixedLimits sind standardmaessig Falsch; RefPerc ist_
↳standardmaessig Wahr
>>> # die Belegung von MCategory gemass FixedLimitsHigh/Low erfolgt immer ...
>>> pFWVB=rm.pltNetDHUS(timeDeltaToT=timeDeltaToT,pFWVBMeasureCBFixedLimitHigh=0.80,
↳pFWVBMeasureCBFixedLimitLow=0.66,pFWVBGCategory=['BLNZ1u5u7'],pVICsDf=pd.DataFrame({
↳'Kundenname': ['VIC1'],'Knotenname': ['V-K007']}))
>>> # ---
>>> # Check pFWVB Return
>>> # ---
>>> f=lambda x: "{0:8.5f}".format(x)
>>> print(pFWVB[['Measure','MCategory','GCategory','VIC']].round(2).to_
↳string(formatters={'Measure':f}))
   Measure MCategory GCategory  VIC
0  0.81000         Top  BLNZ1u5u7  NaN
1  0.67000      Middle         NaN
2  0.66000      Middle  BLNZ1u5u7  NaN
3  0.66000      Bottom  BLNZ1u5u7  VIC1
4  0.69000      Middle         NaN
>>> # ---
>>> # Print
>>> # ---
>>> (wD,fileName)=os.path.split(xm.xmlFile)
>>> (base,ext)=os.path.splitext(fileName)
>>> plotFileName=wD+os.path.sep+base+'.'+'pdf'
>>> if os.path.exists(plotFileName):
...     os.remove(plotFileName)

```

(continues on next page)

(continued from previous page)

```

>>> plt.savefig(plotFileName,dpi=2*dpi_screen)
>>> os.path.exists(plotFileName)
True
>>> # ---
>>> # Plot 3Classes True
>>> # ---
>>> plt.close('all')
>>> # FixedLimits wird automatisch auf Wahr gesetzt wenn 3Classes Wahr ...
>>> pFWVB=rm.pltNetDHUS(timeDeltaToT=timeDeltaToT,pFWVBMeasure3Classes=True,
↳pFWVBMeasureCBFixedLimitHigh=0.80,pFWVBMeasureCBFixedLimitLow=0.66)
>>> # ---
>>> # LocalHeatingNetwork Clean Up
>>> # ---
>>> if os.path.exists(mx.h5File):
...     os.remove(mx.h5File)
>>> if os.path.exists(mx.mxsZipFile):
...     os.remove(mx.mxsZipFile)
>>> if os.path.exists(mx.h5FileVecs):
...     os.remove(mx.h5FileVecs)
>>> if os.path.exists(plotFileName):
...     os.remove(plotFileName)

```

Rm.**pltNetFigAx** (*pDf*, ****kws**)

Erzeugt eine für die Netzdarstellung verzerrungsfreie Axes-Instanz.

- verwendet gcf() (will return an existing figure if one is open, or it will make a new one if there is no active figure)
- an already existing figure might be created this way: fig=plt.figure(dpi=2*72,linewidth=1.)
- errechnet die verzerrungsfreie Darstellung unter Berücksichtigung einer zukünftigen horizontalen Farblegende
- erzeugt eine Axes-Instanz
- setzt Attribute der Axes-Instanz
- setzt Attribute der Figure-Instanz

Args: *pDf*: dataframe

Coordinates:

- *pXCor_i*: colName in *pDf* (default: 'pXCor_i'): x-Start Coordinate of all Edges to be plotted
- *pYCor_i*: colName in *pDf* (default: 'pYCor_i'): y-Start Coordinate of all Edges to be plotted
- *pXCor_k*: colName in *pDf* (default: 'pXCor_k'): x-End Coordinate of all Edges to be plotted
- *pYCor_k*: colName in *pDf* (default: 'pYCor_k'): y-End Coordinate of all Edges to be plotted

Colorlegend:

- *CBFraction*: fraction of original axes to use for colorbar (default: 0.05)
- *CBHpad*: fraction of original axes between colorbar and new image axes (default: 0.0275)

Figure:

- *pltTitle*: title [not suptitle] (default: 'pltNetFigAx')
- *figFrameon*: figure frame (background): displayed or invisible (default: True)
- *figEdgecolor*: edge color of the Figure rectangle (default: 'black')
- *figFacecolor*: face color of the Figure rectangle (default: 'white')

Rm.**pltNetNodes** (*pDf*, ****kws**)

Scatters NODEs on gca().

Args: pDf: dataFrame

NODE: Size (Attribute)

- pAttribute: colName (default: 'Attribute') in pDf
- pSizeFactor: (default: 1.)
- scatter Sy-Area in $pts^2 = pSizeFactor * Attribute$

NODE: Color (Measure)

- pMeasure: colName (default: 'Measure') in pDf
- pMeasureColorMap (default: plt.cm.autumn)
- pMeasureAlpha (default: 0.9)
- pMeasureClip (default: False)
- CBFixedLimits (default: True)
- CBFixedLimitLow (default: 0.)
- CBFixedLimitHigh (default: 1.)

NODE: 3Classes

- pMeasure3Classes (default: True)
- pMCategory: colName (default: 'MCategory') in pDf
- pMCatTopTxt (default: 'Top')
- pMCatMidTxt (default: 'Middle')
- pMCatBotTxt (default: 'Bottom')
- pMCatTopColor (default: 'palegreen')
- pMCatTopAlpha (default: 0.9)
- pMCatTopClip (default: False)
- pMCatMidColorMap (default: plt.cm.autumn)
- pMCatMidAlpha (default: 0.9)
- pMCatMidClip (default: False)
- pMCatBotColor (default: 'violet')
- pMCatBotAlpha (default: 0.9)
- pMCatBotClip (default: False)

NODE:

- pXCor: colName (default: 'pXCor_i') in pDf
- pYCor: colName (default: 'pYCor_i') in pDf

Returns: (pcN, vmin, vmax)

- pcN: die mit Farbskala gezeichneten Symbole
- vmin/vmax: die für die Farbskala verwendeten Extremalwerte

Rm.**pltNetPipes** (pDf, ***kws*)

Plots Lines with Marker on gca().

Args: pDf: dataFrame

PIPE-Line:

- pAttribute: column in pDf (default: 'Attribute')

- `pAttributeLs` (default: `'-'`)
- `pAttributeSizeFactor`: plot linewidth in pts = `pAttributeSizeFactor` (default: 1.0) * Attribute
- `pAttributeColorMap` (default: `plt.cm.binary`)
- `pAttributeColorMapUsageStart` (default: 1./3; Wertebereich: [0,1])
 - Farbskala nach vorh. min./max. Wert
 - die Farbskala wird nur ab UsageStart genutzt
 - d.h. Werte die eine “kleinere” Farbe hätten, bekommen die Farbe von UsageStart

PIPE-Marker:

- `pMeasure`: column in `pDf` (default: `'Measure'`)
- `pMeasureMarker` (default: `'.'`)
- `pMeasureSizeFactor`: plot markersize in pts = `pMeasureSizeFactor` (default: 1.0) * Measure
- `pMeasureColorMap` (default: `plt.cm.cool`)
- `pMeasureColorMapUsageStart` (default: 0.; Wertebereich: [0,1])
 - Farbskala nach vorh. min./max. Wert
 - die Farbskala wird nur ab UsageStart genutzt
 - d.h. Werte die eine “kleinere” Farbe hätten, bekommen die Farbe von UsageStart

PIPE:

- `pWAYPXCors`: column in `pDf` (default: `'pWAYPXCors'`)
- `pWAYPYCors`: column in `pDf` (default: `'pWAYPYCors'`)
- `pClip` (default: `False`)

Rm. **`pltNetLegendColorbar`** (*pc, pDf, **kws*)

Erzeugt eine Axes `cax` für den Legendenbereich aus `ax` (`=gca()`) und zeichnet auf `cax` die Farblegende (die Farbskala mit allen Eigenschaften).

Args: `pc`: (eingefärbte) `PathCollection` (aus `pltNetNodes`); wird für die Erzeugung der Farbskala zwingend benötigt `pDf`: `dataFrame` (default: `None`)

Measure:

- `pMeasure`: `colName` in `pDf` (default: `'Measure'`)
- `pMeasureInPerc`: Measure wird interpretiert in Prozent [0-1] (default: `True`)
- `pMeasure3Classes` (default: `False` d.h. Measure wird nicht in 3 Klassen dargestellt)

CBFixedLimits (Ticks):

- `CBFixedLimits` (default: `False` d.h. Farbskala nach vorh. min./max. Wert)
- `CBFixedLimitLow` (default: `.10`)
- `CBFixedLimitHigh` (default: `.95`)

Label:

- `pMeasureUNIT` (default: `'[]'`)
- `pMeasureTYPE` (default: `''`)

CB

- CBFraction: fraction of original axes to use for colorbar (default: 0.05)
- CBHpad: fraction of original axes between colorbar and new image axes (default: 0.0275)
- CBLLabelPad (default: -50)
- CBTicklabelsHPad (default: 0.)
- CBAspect: ratio of long to short dimension (default: 10.)
- CBSShrink: fraction by which to shrink the colorbar (default: 0.3)
- CBAnchorHorizontal: horizontaler Fußpunkt der colorbar in Plot-% (default: 0.)
- CBAnchorVertical: vertikaler Fußpunkt der colorbar in Plot-% (default: 0.2)

Return: cax

Rm.**pltNetLegendColorbar3Classes** (*pDf*, ***kws*)

Zeichnet auf gca() die ergaenzenden Legendeninformationen bei 3 Klassen.

- scatters the Top-Symbol
- scatters the Bot-Symbol
- the “Mid-Symbol” is the (already existing) colorbar with (already existing) ticks and ticklabels

Args:

pDf: dataframe

Category:

- pMCategory: colName in pDf (default: ‘MCategory’)
- pMCatTopText
- pMCatMidText
- pMCatBotText

CBLegend (3Classes) - Parameterization of the representative Symbols

- CBLe3cTopVPad (default: 1+1*1/4)
- CBLe3cMidVPad (default: .5)
- CBLe3cBotVPad (default: 0-1*1/4)
 - 1 is the height of the Colorbar
 - the VPads (the vertical Sy-Positions) are defined in cax.transAxes Coordinates
 - cax is the Colorbar Axes
- CBLe3cSySize=10**2 (Sy-Area in pts^2)
- CBLe3cSyType=’o’

Color:

- pMCatBotColor=’violet’
- pMCatTopColor=’palegreen’

Returns: (bbTop, bbMid, bbBot): the boundingBoxes of the 3Classes-Symbols

class Rm.**Rm** (*xm=None*, *mx=None*)

classmethod **pltNetPipes** (*pDf*, ***kws*)

Plots colored PIPES.

Args:

DATA:

pDf: dataframe

- query: query to filter pDf; default: None; Exp.: `"CONT_ID == '1001'"`
- fmask: function to filter pDf; default: None; Exp.: `=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2' else False`
- query and fmask are used both (query 1st) if not None
- sort_values_by: list of colNames defining the plot order; default: None (d.h. die Plotreihenfolge - und damit die z-Order - ist dann die pDF-Reihenfolge)
- sort_values_ascending; default: False (d.h. kleine zuletzt und damit (wenn pAttrLineSize = pAttribute/pAttributeFunc) auch dünne über dicke); nur relevant bei sort_values_by

AXES: pAx: Axes to be plotted on; if not specified: `gca()` is used

Colorlegend:

- CBFraction in % (default: 5)
- CBHpad (default: 0.05)
- CBLabel (default: pAttribute/pAttributeFunc)
- CBBinTicks (default: None, d.h. keine Vorgabe von Außen); Vorgabe N: N yTicks; bei diskreten CM gemeint im Sinne von N-1 diskreten Kategorien
- CBBinDiscrete (default: False, d.h. eine gegebene (kontinuierliche) CM wird nicht in eine diskrete gewandelt)
- wenn CBBinDiscrete, dann gilt N aus CBBinTicks fuer die Ticks (bzw. Kategorien); ist CBBinTicks undef. gilt 4 (also 3 Kategorien)
- bei den vorgenannten Kategorien handelt es sich um eine gleichmäßige Unterteilung des definierten Wertebereiches
- CBBinBounds (default: None): wenn die CM eine diskrete ist, dann wird eine vorgegebene BoundaryNorm angewandt; CBBinTicks hat dann keine Bedeutung
- CBTicks: individuell vorgegebene Ticks; wird am Schluss prozessiert, d.h. vorh. (ggf. auch durch CBBinTicks bzw. `<=/>=` u. `v=/^=` bereits manipulierte) ...
- ... Ticks werden überschrieben; kann ohne CBTickLabels verwendet werden
- CBTickLabels: individuell vorgegebene Ticklabels; wird danach prozessiert; Länge muss zu dann existierenden Ticks passen; kann auch ohne CBTicks verwendet werden

PIPE-Attribute:

- pAttribute: column in pDf (default: 'Attribute')
- pAttributeFunc:
 - function to be used to construct a new col to be plotted
 - if pAttributeFunc is not None pAttribute is not used: pAttribute is set to 'pAttributeFunc'
 - the new constructed col is named 'pAttributeFunc'; this name can be used in sort_values_by

PIPE-Color:

- pAttributeColorMap (default: plt.cm.cool)

- Farbskalamapping:
-
- `pAttributeColorMapMin` (default: `pAttribute.min()`); ordnet der kleinsten Farbe einen Wert zu; CM: wenn angegeben _und unterschritten: `<=`
- `pAttributeColorMapMax` (default: `pAttribute.max()`); ordnet der größten Farbe einen Wert zu; CM: wenn angegeben _und überschritten: `>=`
- Standard: Farbskala wird voll ausgenutzt; d.h. der (ggf. mit Min/Max) eingegrenzte Wertebereich wird den Randfarben der Skala zugeordnet
- wenn ein anderer, kleinerer, Wertebereich mit derselben Farbskala geplottet wird, dann sind die Farben in den Plots nicht vergleichbar ...
- ... wenn eine Farbvergleichbarkeit erzielt werden soll, darf dieselbe Farbskala nicht voll ausgenutzt werden
- **`pAttributeColorMapUsageStart` (default: 0.; Wertebereich: [0,1[)**
 - hier: die Farbskala wird unten nur ab `UsageStart` genutzt ...
 - ... d.h. Werte die eine “kleinere” Farbe hätten, bekommen die Farbe von `UsageStart`; CM: `v=`
- **`pAttributeColorMapUsageEnd` (default: 1.; Wertebereich:]0,1])**
 - hier: die Farbskala wird oben nur bis `UsageEnd` genutzt ...
 - ... d.h. Werte die eine “größere” Farbe hätten, bekommen die Farbe von `UsageEnd`; CM: `^=`
- etwas anderes ist es, wenn man eine Farbskala an den Rändern nicht voll ausnutzen möchte weil einem die Farben dort nicht gefallen ...

PIPE-Color 2nd:

- um “unwichtige” Bereiche zu “dimmen”: Beispiele:
- räumlich: nicht-Schnitt Bereiche; Bestand (2nd) vs. Ausbau; Zonen unwichtig (2nd) vs. Zonen wichtig; Ok (2nd) von NOK
- es werden erst die 2nd-Color Pipes gezeichnet; die (1st-)Color Pipes werden danach gezeichnet, liegen also “über” den “unwichtigen”
- es wird dieselbe Spalte `pAttribute/pAttributeFunc` für die 2. Farbskala verwendet
- es wird derselbe Linienstil (`pAttributeLs`) für die 2. Farbskala verwendet
- es wird dieselbe Dicke `pAttrLineSize` (`pAttribute/pAttributeFunc`) für die 2. Farbskala verwendet
- nur die Farbskala ist anders sowie ggf. das Farbskalamapping
- `pAttributeColorMapFmask`: function to filter `pDf` to decide to plot with `colorMap`; default: `=lambda row: True`
- `pAttributeColorMap2ndFmask`: function to filter `pDf` to decide to plot with `colorMap2nd`; default: `=lambda row: False`
- mit den beiden Funktionsmasken kann eine Filterung zusätzlich zu `query` und `fmask` realisiert werden
- die Funktionsmasken sollten schnittmengenfrei sein; wenn nicht: 2nd überschreibt
- `pAttributeColorMap2nd` (default: `plt.cm.binary`)

- Farbskalamapping:
-
- pAttributeColorMap2ndMin (default: pAttributeColorMapMin)
- pAttributeColorMap2ndMax (default: pAttributeColorMapMax)
- die Farbskala wird an den Rändern nicht voll ausgenutzt wenn die Farben dort ggf. nicht gefallen:
- pAttributeColorMap2ndUsageStart (default: 0.; Wertebereich: [0,1[)
- pAttributeColorMap2ndUsageEnd (default: 1.; Wertebereich:]0,1])

PIPE-Linestyle:

- pAttributeLs (default: '-')
- same for all colors if mutiple colors are specified

PIPE-Linesize:

- pAttrLineSize: column in pDf; if not specified: pAttribute/pAttributeFunc
- pAttrLineSizeFactor (>0): plot linewidth in pts = pAttrLineSizeFactor (default: =...) * fabs(pAttrLineSize)
- ...: 1./(pDf[pAttrLineSize].std()*2.)
- same for all colors if mutiple colors are specified

PIPE-Geometry:

- pWAYPXCors: column in pDf (default: 'pWAYPXCors')
- pWAYPYCors: column in pDf (default: 'pWAYPYCors')
- pClip (default: True)

```
>>> import pandas as pd
>>> import matplotlib
>>> import matplotlib.pyplot as plt
>>> import matplotlib.gridspec as gridspec
>>> import math
>>> # ---
>>> try:
...     import Rm
... except ImportError:
...     from PT3S import Rm
>>> # ---
>>> xm=xms['DHNetwork']
>>> #mx=mxs['DHNetwork']
>>> # ---
>>> plt.close()
>>> size_DINA3quer=(16.5, 11.7)
>>> dpiSize=72
>>> fig=plt.figure(figsize=size_DINA3quer,dpi=dpiSize)
>>> gs = gridspec.GridSpec(4, 2)
>>> # ---
>>> vROHR=xm.dataFrames['vROHR']
>>> # ---
>>> # Attribute (with neg. Values)
>>> # -----
```

(continues on next page)

(continued from previous page)

```

>>> axNfd = fig.add_subplot(gs[0])
>>> Rm.Rm.pltNetPipes(vROHR
...     ,query="CONT_ID == '1001'"
...     ,fmask=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2'
↳else False
...     ,pAx=axNfd
...     ,pAttribute='ROHR~*~*~*~QMAV'
...     )
>>> txt=axNfd.set_title('RL QMAV')
>>> # ---
>>> # Function as Attribute
>>> # -----
>>> axNfd = fig.add_subplot(gs[1])
>>> Rm.Rm.pltNetPipes(vROHR
...     ,query="CONT_ID == '1001'"
...     ,fmask=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2'
↳else False
...     ,pAx=axNfd
...     ,pAttributeFunc=lambda row: math.fabs(row['ROHR~*~*~*~QMAV'])
...     )
>>> txt=axNfd.set_title('RL QMAV Abs')
>>> # -----
>>> # ---
>>> # Mi/MaD zS auf
>>> # -----
>>> axNfd = fig.add_subplot(gs[2])
>>> Rm.Rm.pltNetPipes(vROHR
...     ,query="CONT_ID == '1001'"
...     ,fmask=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2'
↳else False
...     ,pAx=axNfd
...     ,pAttributeFunc=lambda row: math.fabs(row['ROHR~*~*~*~QMAV'])
...     ,pAttributeColorMapMin=0.
...     ,pAttributeColorMapMax=1600.
...     ,CBLLabel='Q [t/h]'
...     ,sort_values_by=['pAttributeFunc']
...     ,sort_values_ascending=True
...     )
>>> txt=axNfd.set_title('Mi/MaD zS auf')
>>> # -----
>>> # ---
>>> # ind. Kategorien
>>> # -----
>>> axNfd = fig.add_subplot(gs[3])
>>> cm = matplotlib.colors.ListedColormap(['cyan', 'royalblue', 'magenta
↳', 'coral'])
>>> cm.set_over('0.25')
>>> cm.set_under('0.75')
>>> bounds = [10.,100.,200.,800.,1600.]
>>> norm = matplotlib.colors.BoundaryNorm(bounds, cm.N)
>>> Rm.Rm.pltNetPipes(vROHR
...     ,query="CONT_ID == '1001'"
...     ,fmask=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2'
↳else False
...     ,pAx=axNfd
...     ,pAttributeFunc=lambda row: math.fabs(row['ROHR~*~*~*~QMAV'])
...     ,pAttributeColorMap=cm

```

(continues on next page)

(continued from previous page)

```

...     ,CBBinBounds=bounds
...     ,CBLLabel='Q [t/h]'
...     ,sort_values_by=['pAttributeFunc']
...     ,sort_values_ascending=True
... )
>>> txt=axNfd.set_title('ind. Kategorien')
>>> # -----
>>> # ---
>>> # Unwichtiges ausblenden über 2nd Color
>>> # -----
>>> vAGSN=xm.dataFrames['vAGSN']
>>> hpRL=vAGSN[(vAGSN['LFDNR']=='1') & (vAGSN['Layer']==2)]
>>> pDf=pd.merge(vROHR
...     ,hpRL[hpRL.IptIdx=='S'] # wg. Innenpunkte
...     ,how='left'
...     ,left_on='pk'
...     ,right_on='OBJID'
...     ,suffixes=(',', '_AGSN')).filter(items=vROHR.columns.tolist()+[
↳ 'OBJID'])
>>> axNfd = fig.add_subplot(gs[4])
>>> Rm.Rm.pltNetPipes(pDf
...     ,query="CONT_ID == '1001'"
...     ,fmask=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2'
↳ else False
...     ,pAx=axNfd
...     ,pAttributeFunc=lambda row: math.fabs(row['ROHR~*~*~*~QMAV'])
...     ,pAttributeColorMapMin=0.
...     ,pAttributeColorMapMax=1500.
...     ,CBBinTicks=7
...     ,CBLLabel='Q [t/h]'
...     ,sort_values_by=['pAttributeFunc']
...     ,sort_values_ascending=True
...     ,pAttributeColorMapFmask=lambda row: True if not pd.isnull(row.
↳ OBJID) else False
...     ,pAttributeColorMap2ndFmask=lambda row: True if pd.isnull(row.
↳ OBJID) else False
... )
>>> txt=axNfd.set_title('Unwichtiges ausblenden über 2nd Color')
>>> # -----
>>> # ---
>>> # Farbskalen an den Rändern abschneiden
>>> # -----
>>> axNfd = fig.add_subplot(gs[5])
>>> Rm.Rm.pltNetPipes(pDf
...     ,query="CONT_ID == '1001'"
...     ,fmask=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2'
↳ else False
...     ,pAx=axNfd
...     ,pAttributeFunc=lambda row: math.fabs(row['ROHR~*~*~*~QMAV'])
...     ,pAttributeColorMapMin=0.
...     ,pAttributeColorMapMax=1500.
...     ,CBLLabel='Q [t/h]'
...     ,sort_values_by=['pAttributeFunc']
...     ,sort_values_ascending=True
...     ,pAttributeColorMapFmask=lambda row: True if not pd.isnull(row.
↳ OBJID) else False
...     ,pAttributeColorMap2ndFmask=lambda row: True if pd.isnull(row.
↳ OBJID) else False

```

(continues on next page)

(continued from previous page)

```

...     ,pAttributeColorMap2ndUsageStart=.5/5. # nicht zu weiß
...     ,pAttributeColorMap2ndUsageEnd=2.5/5. # nicht zu schwarz
...     ,pAttributeColorMapUsageStart=3/15.
...     ,pAttributeColorMapUsageEnd=12/15.
... )
>>> txt=axNfd.set_title('Farbskalen an den Rändern abschneiden')
>>> # -----
>>> # ---
>>> # Farbskala diskretisieren
>>> # -----
>>> axNfd = fig.add_subplot(gs[6])
>>> Rm.Rm.pltNetPipes(pDf
...     ,query="CONT_ID == '1001'"
...     ,fmask=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2'
↳ else False
...     ,pAx=axNfd
...     ,pAttributeFunc=lambda row: math.fabs(row['ROHR~*~*~*~QMAV'])
...     ,pAttributeColorMapMin=0.
...     ,pAttributeColorMapMax=1500.
...     ,CBBinDiscrete=True
...     ,CBLLabel='Q [t/h]'
...     ,sort_values_by=['pAttributeFunc']
...     ,sort_values_ascending=True
...     ,pAttributeColorMapFmask=lambda row: True if not pd.isnull(row.
↳ OBJID) else False
...     ,pAttributeColorMap2ndFmask=lambda row: True if pd.isnull(row.
↳ OBJID) else False
...     ,pAttributeColorMap2ndUsageStart=.5/5. # nicht zu weiß
...     ,pAttributeColorMap2ndUsageEnd=2.5/5. # nicht zu schwarz
...     ,CBTicks=[250,750,1250]
...     ,CBTickLabels=['klein','mittel','groß']
... )
>>> txt=axNfd.set_title('Farbskala diskretisieren')
>>> # -----
>>> # ---
>>> # Unterkategorien
>>> # -----
>>> baseColorsDef="tab10"
>>> catagoryColors=[9,6,1]
>>> nOfSubCatsReq=4
>>> cm=Rm.pltMakeCategoricalCmap(baseColorsDef=baseColorsDef,
↳ catagoryColors=catagoryColors,nOfSubCatsReq=nOfSubCatsReq,
↳ reversedSubCatOrder=True)
>>> axNfd = fig.add_subplot(gs[7])
>>> Rm.Rm.pltNetPipes(pDf
...     ,query="CONT_ID == '1001'"
...     ,fmask=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2'
↳ else False
...     ,pAx=axNfd
...     ,pAttributeFunc=lambda row: math.fabs(row['ROHR~*~*~*~QMAV'])
...     ,pAttributeColorMap=cm
...     ,pAttributeColorMapMin=0.
...     ,pAttributeColorMapMax=1500.
...     ,CBBinTicks=16
...     ,CBLLabel='Q [t/h]'
...     ,sort_values_by=['pAttributeFunc']
...     ,sort_values_ascending=True

```

(continues on next page)

(continued from previous page)

```

...     ,pAttributeColorMapFmask=lambda row: True if not pd.isnull(row.
↳OBJID) else False
...     ,pAttributeColorMap2ndFmask=lambda row: True if pd.isnull(row.
↳OBJID) else False
...     ,pAttributeColorMap2ndUsageStart=.5/5. # nicht zu weiß
...     ,pAttributeColorMap2ndUsageEnd=2.5/5. # nicht zu schwarz
↳
↳
...     )
>>> txt=axNfd.set_title('Unterkategorien')
>>> # -----
>>> gs.tight_layout(fig)
>>> plt.show()
>>> plt.savefig('pltNetPipes.pdf',format='pdf',dpi=dpiSize*2)
>>> # -----
>>> plt.close()
>>> fig=plt.figure(figsize=Rm.DINA3q,dpi=dpiSize)
>>> gs = gridspec.GridSpec(1, 1)
>>> # ---
>>> #
>>> # -----
>>> axNfd = fig.add_subplot(gs[0])
>>> Rm.Rm.pltNetPipes(vROHR
...     ,query="CONT_ID == '1001'"
...     ,fmask=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2' and
↳row.LTGR_NAME=='NWDUF2' else False
...     ,pAx=axNfd
...     ,pAttributeFunc=lambda row: math.fabs(row['ROHR~*~*~*~QMAV'])
...     )
>>> txt=axNfd.set_title('RL QMAV Abs (Ausschnitt)')
>>> gs.tight_layout(fig)
>>> plt.show()
>>> # -----
>>> plt.close()
>>> fig=plt.figure(figsize=Rm.DINA3,dpi=dpiSize)
>>> gs = gridspec.GridSpec(1, 1)
>>> # ---
>>> #
>>> # -----
>>> axNfd = fig.add_subplot(gs[0])
>>> Rm.Rm.pltNetPipes(vROHR
...     ,query="CONT_ID == '1001'"
...     ,fmask=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2' and
↳row.LTGR_NAME=='NWDUF2' else False
...     ,pAx=axNfd
...     ,pAttributeFunc=lambda row: math.fabs(row['ROHR~*~*~*~QMAV'])
...     )
>>> txt=axNfd.set_title('RL QMAV Abs (Ausschnitt)')
>>> gs.tight_layout(fig)
>>> plt.show()

```

classmethod **pltHP** (*pDf*, ****kws**)

Plots a Hydraulic Profile.

Args:

DATA: *pDf*: *dataFrame*

defining the HPLINES (xy-curves) Identification:

the different HPs in pDf are identified by the two cols NAMECol: default: 'NAME'; set to None if NAMECol is not criteria for Identification ... and LayerCol: default: 'Layer'; set to None if LayerCol is not criteria for Identification ...

for each HP several lines (xy-curves) are plotted

... not criteria ... if NAMECol is None only LayerCol is used if LayerCol also is None, all rows are treated as "the" HPLINE

defining the HPLINES (xy-curves) Geometry:

- xCol: col in pDf for x; example: 'x'

the col is the same for all HPs and all y

- edgeNodesColSequence: cols to be used for start-node, end-node, next-node; default: ['NAME_i', 'NAME_k', 'nextNODE']
- 'NAME'_'Layer' (i.e. Nord-Süd_1) NAMECol_LayerCol is used as an Index in hpLineGeoms
- hpLineGeoms - Example - = {

```
'V-Abzweig_1':{ 'masterHP': 'AGFW Symposium DH_1', 'masterNode': 'V-3107', 'matchType': 'starts' } } - masterHP: Bezugs-Schnitt - masterNode: muss es in masterHP geben - masterNode: muss es auch im Schnitt geben bei matchType='matches'; bei 'starts' wird der Anfang gemapped; bei 'ends' das Ende
```

defining the HPLINES (xy-curves) y-Achsentypen (y-Axes):

- hpLines: list of cols in pDf for y; example: ['P']

each col in hpLines defines a hpLine (a xy-curve) to be plotted for each identified HP
all defined hpLines are plotted

defining the HPLINES (xy-curves) Layout:

'NAME'_'Layer'_'hpLineType' (i.e. Nord-Süd_1_P) is used as an Index in hpLine-Props

- **hpLineProps - Example - = {**
 'Nord-Süd_1_P':{ 'label': 'VL', 'color': 'red', 'linestyle': '-', 'linewidth': 3 }
 'Nord-Süd_2_P':{ 'label': 'RL', 'color': 'blue', 'linestyle': '-', 'linewidth': 3 }
}

if 'NAME'_'Layer'_'hpLine' not in hpLineProps: default props are used

if hpLineProps['NAME'_'Layer'_'hpLine'] == None: HPLINE is not plotted

y-Achsentypen (y-Axes):

- werden ermittelt aus hpLines
- der Spaltenname - z.B. 'P' - wird dabei als Bezeichner für den Achsentyp benutzt
- die Achsen werden erstellt in der Reihenfolge in der sie in hpLines auftreten
- Bezeichner wie 'P', 'P_1', ... werden dabei als vom selben Achsentyp 'P' (selbe y-Achse also) gewertet

- P_1, P_2, ... können z.B. P zu verschiedenen Zeiten sein oder Aggregate über die Zeit wie Min/Max
- yAxesDetectionPattern: regExp mit welcher die Achsentypen ermittelt werden; default: '([w]+)(_)(d+)\$'
- yTwinedAxesPosDeltaHPStart: (i.d.R. negativer) Abstand der 1. y-Achse von der Zeichenfläche; default: -0.0125
- yTwinedAxesPosDeltaHP: (i.d.R. negativer) zus. Abstand jeder weiteren y-Achse von der Zeichenfläche; default: -0.05

AXES: pAx: Axes to be plotted on; if not specified: gca() is used

Return: yAxes: dct with AXES; key=y-Achsentypen yLines: dct with Line2Ds; key=Index from hpLineProps xNodeInfs: dct with NodeInformation; key=Index also used in i.e. hpLineGeoms

key: NAMECol_LayerCol value: dct

key: node value: dct

kwds['xCol']: x in HP kwds['xCol']+'Plot': x in HP-Plot pDfIdx:
Index in pDf

```
>>> # -q -m 0 -s pltHP -y no -z no -w DHNetwork
>>> import pandas as pd
>>> import matplotlib
>>> import matplotlib.pyplot as plt
>>> import matplotlib.gridspec as gridspec
>>> import math
>>> try:
...     import Rm
... except ImportError:
...     from PT3S import Rm
>>> # ---
>>> xm=xms['DHNetwork']
>>> mx=mxs['DHNetwork']
>>> xm.MxAdd(mx=mx, aggReq=['TIME', 'TMIN', 'TMAX'], timeReq=3*[mx.df.
↪ index[0]], timeReq2nd=3*[mx.df.index[-1]], viewList=['vAGSN'],
↪ ForceNoH5Update=True)
>>> vAGSN=xm.dataFrames['vAGSN']
>>> for PH,P,RHO,Z in zip(['PH', 'PH_1', 'PH_2'], ['P', 'P_1', 'P_2'], ['RHO',
↪ 'RHO_1', 'RHO_2'], ['Z', 'Z_1', 'Z_2']):
...     vAGSN[PH]=vAGSN.apply(lambda row: row[P]*math.pow(10.,5.)/
↪ (row[RHO]*9.81), axis=1)
...     vAGSN[PH]=vAGSN[PH]+vAGSN[Z].astype('float64')
>>> for bBzg,P,RHO,Z in zip(['bBzg', 'bBzg_1', 'bBzg_2'], ['P', 'P_1', 'P_2'],
↪ ['RHO', 'RHO_1', 'RHO_2'], ['Z', 'Z_1', 'Z_2']):
...     vAGSN[bBzg]=vAGSN.apply(lambda row: row[RHO]*9.81/math.pow(10.,5.
↪ ), axis=1)
...     vAGSN[bBzg]=vAGSN[P]+vAGSN[Z].astype('float64')*vAGSN[bBzg]
>>> plt.close()
>>> fig=plt.figure(figsize=Rm.DINA3q, dpi=Rm.dpiSize)
>>> gs = gridspec.GridSpec(3, 1)
>>> # -----
>>> axNfd = fig.add_subplot(gs[0])
>>> yAxes,yLines,xNodeInfs=Rm.Rm.pltHP(vAGSN,pAx=axNfd
... ,hpLines=['bBzg', 'bBzg_1', 'bBzg_2', 'Q']
... ,hpLineProps={
...     'AGFW Symposium DH_1_bBzg':{'label':'VL', 'color':'red' ,
↪ 'linestyle':'--', 'linewidth':3}
```

(continues on next page)

(continued from previous page)

```

...     , 'AGFW Symposium DH_2_bBzg': {'label': 'RL', 'color': 'blue',
↳ 'linestyle': '-', 'linewidth': 3}
...     , 'AGFW Symposium DH_2_bBzg_1': {'label': 'RL min', 'color': 'blue',
↳ 'linestyle': '-.', 'linewidth': 1}
...     , 'AGFW Symposium DH_1_bBzg_2': {'label': 'VL max', 'color': 'red' ,
↳ 'linestyle': '-.', 'linewidth': 1}
...     , 'AGFW Symposium DH_1_bBzg_1': None
...     , 'AGFW Symposium DH_2_bBzg_2': None
...     , 'AGFW Symposium DH_1_Q': {'label': 'VL Q', 'color': 'magenta' ,
↳ 'linestyle': '--', 'linewidth': 2}
...     , 'AGFW Symposium DH_2_Q': {'label': 'RL Q', 'color': 'lightblue',
↳ 'linestyle': '--', 'linewidth': 2}
... }
... )
>>> yAxes.keys()
dict_keys(['bBzg', 'Q'])
>>> yLines.keys()
dict_keys(['AGFW Symposium DH_1_bBzg', 'AGFW Symposium DH_1_bBzg_2',
↳ 'AGFW Symposium DH_1_Q', 'AGFW Symposium DH_2_bBzg', 'AGFW Symposium_
↳ DH_2_bBzg_1', 'AGFW Symposium DH_2_Q'])
>>> txt=axNfd.set_title('HP')
>>> gs.tight_layout(fig)
>>> plt.show()
>>> ###
>>> Rcuts=[
...     {'NAME': 'R-Abzweig', 'nl': ['R-3107', 'R-3427']}
...     , {'NAME': 'R-EndsTest', 'nl': ['R-HWSU', 'R-HKW3S']}
...     , {'NAME': 'R-MatchesTest', 'nl': ['R-HKW1', 'R-2104']}
... ]
>>> Vcuts=[
...     {'NAME': 'V-Abzweig', 'nl': ['V-3107', 'V-3427']}
...     , {'NAME': 'V-EndsTest', 'nl': ['V-HWSU', 'V-HKW3S']}
...     , {'NAME': 'V-MatchesTest', 'nl': ['V-HKW1', 'V-2104']}
... ]
>>> fV=lambda row: True if row.KVR_i=='1' and row.KVR_k=='1' else False
>>> fR=lambda row: True if row.KVR_i=='2' and row.KVR_k=='2' else False
>>> for vcut,rcut in zip(Vcuts,Rcuts):
...     ret=xm.vAGSN_Add(nl=vcut['nl'],weight='L',Layer=1,AKTIV=None,
↳ NAME=vcut['NAME'],fmask=fV)
...     ret=xm.vAGSN_Add(nl=rcut['nl'],weight='L',Layer=2,AKTIV=None,
↳ NAME=rcut['NAME'],fmask=fR)
>>> # Schnitte erneut mit Ergebnissen versorgen, da Schnitte neu_
↳ definiert wurden
>>> xm.MxAdd(mx=mx,ForceNoH5Update=True)
>>> vAGSN=xm.dataFrames['vAGSN']
>>> for PH,P,RHO,Z in zip(['PH'], ['P'], ['RHO'], ['Z']):
...     vAGSN[PH]=vAGSN.apply(lambda row: row[P]*math.pow(10.,5.) /
↳ (row[RHO]*9.81),axis=1)
...     vAGSN[PH]=vAGSN[PH]+vAGSN[Z].astype('float64')
>>> for bBzg,P,RHO,Z in zip(['bBzg'], ['P'], ['RHO'], ['Z']):
...     vAGSN[bBzg]=vAGSN.apply(lambda row: row[RHO]*9.81/math.pow(10.,5.
↳ ),axis=1)
...     vAGSN[bBzg]=vAGSN[P]+vAGSN[Z].astype('float64')*vAGSN[bBzg]
>>> plt.close()
>>> fig=plt.figure(figsize=Rm.DINA3q,dpi=Rm.dpiSize)
>>> gs = gridspec.GridSpec(3, 1)
>>> # -----

```

(continues on next page)

(continued from previous page)

```

>>> axNfd = fig.add_subplot(gs[0])
>>> yAxes,yLines,xNodeInfs=Rm.Rm.pltHP(vAGSN[vAGSN['NAME'].isin(['R-
↳ Abzweig','V-Abzweig','AGFW Symposium DH','R-EndsTest','V-EndsTest','R-
↳ MatchesTest','V-MatchesTest'])],pAx=axNfd
... ,hpLines=['bBzg','Q']
... ,hpLineGeoms={
...     'V-Abzweig_1':{'masterHP':'AGFW Symposium DH_1','masterNode':'V-
↳ 3107','matchType':'starts'}
...     , 'R-Abzweig_2':{'masterHP':'AGFW Symposium DH_2','masterNode':'R-
↳ 3107','matchType':'starts'}
...     , 'V-EndsTest_1':{'masterHP':'AGFW Symposium DH_1','masterNode':'V-
↳ HKW3S','matchType':'ends'}
...     , 'R-EndsTest_2':{'masterHP':'AGFW Symposium DH_2','masterNode':'R-
↳ HKW3S','matchType':'ends'}
...     , 'V-MatchesTest_1':{'masterHP':'AGFW Symposium DH_1','masterNode':
↳ 'V-1312','matchType':'matches','offset':-500}
...     , 'R-MatchesTest_2':{'masterHP':'AGFW Symposium DH_2','masterNode':
↳ 'R-1312','matchType':'matches'}
... }
... ,hpLineProps={
...     'AGFW Symposium DH_1_bBzg':{'label':'VL','color':'red' ,
↳ 'linestyle':'-','linewidth':3}
...     , 'AGFW Symposium DH_2_bBzg':{'label':'RL','color':'blue',
↳ 'linestyle':'-','linewidth':3}
...     , 'AGFW Symposium DH_1_Q':{'label':'VL Q','color':'magenta' ,
↳ 'linestyle':'--','linewidth':2}
...     , 'AGFW Symposium DH_2_Q':{'label':'RL Q','color':'lightblue',
↳ 'linestyle':'--','linewidth':2}
...     , 'V-Abzweig_1_bBzg':{'label':'VL','color':'tomato' , 'linestyle':'-
↳ ', 'linewidth':3}
...     , 'R-Abzweig_2_bBzg':{'label':'RL','color':'plum' , 'linestyle':'-',
↳ 'linewidth':3}
...     , 'V-Abzweig_1_Q':{'label':'VL Q','color':'magenta' , 'linestyle':'-
↳ ', 'linewidth':2}
...     , 'R-Abzweig_2_Q':{'label':'VL Q','color':'lightblue' , 'linestyle':
↳ '--', 'linewidth':2}
...     , 'V-EndsTest_1_bBzg':{'label':'VL','color':'lightcoral' ,
↳ 'linestyle':'-','linewidth':3}
...     , 'R-EndsTest_2_bBzg':{'label':'RL','color':'aquamarine' ,
↳ 'linestyle':'-','linewidth':3}
...     , 'V-EndsTest_1_Q':{'label':'VL Q','color':'magenta' , 'linestyle':
↳ '--', 'linewidth':2}
...     , 'R-EndsTest_2_Q':{'label':'VL Q','color':'lightblue' , 'linestyle
↳ ': '--', 'linewidth':2}
...     , #, 'V-MatchesTest_1_bBzg':{'label':'VL','color':'orange' ,
↳ 'linestyle':'-','linewidth':1}
...     , 'R-MatchesTest_2_bBzg':{'label':'RL','color':'slateblue' ,
↳ 'linestyle':'-','linewidth':1}
...     , 'V-MatchesTest_1_Q':{'label':'VL Q','color':'magenta' , 'linestyle
↳ ': '--', 'linewidth':2}
...     , 'R-MatchesTest_2_Q':{'label':'VL Q','color':'lightblue' ,
↳ 'linestyle':'--', 'linewidth':2}
... }
... )
>>> txt=axNfd.set_title('HP')
>>> gs.tight_layout(fig)
>>> plt.show()

```

(continues on next page)

(continued from previous page)

```

>>> sorted(xNodeInfs.keys())
['AGFW Symposium DH_1', 'AGFW Symposium DH_2', 'R-Abzweig_2', 'R-
↳EndsTest_2', 'R-MatchesTest_2', 'V-Abzweig_1', 'V-EndsTest_1', 'V-
↳MatchesTest_1']
>>> xNodeInf=xNodeInfs['R-Abzweig_2']
>>> nl=Rcuts[0]['nl']
>>> nodeInfS=xNodeInf[nl[0]]
>>> nodeInfE=xNodeInf[nl[-1]]
>>> sorted(nodeInfS.keys())
['pDfIdx', 'x', 'xPlot']
>>> dxPlot=nodeInfE['xPlot']-nodeInfS['xPlot']
>>> dxHP=nodeInfE['x']-nodeInfS['x']
>>> dxPlot==dxHP
True
>>> nodeInfE['x']=round(nodeInfE['x'],3)
>>> nodeInfE['xPlot']=round(nodeInfE['xPlot'],3)
>>> {key:value for key,value in nodeInfE.items() if key not in ['pDfIdx
↳']}
{'x': 3285.0, 'xPlot': 20312.428}
>>>

```

classmethod **pltTC** (*pDf*, *tcLines*, ***kws*)

Plots a Time Curve Diagram.

Args:

DATA:

pDf: **dataFrame** index: times cols: values (with mx.df colnames)

tcLines: dct

defining the Curves and their Layout:

Key: OBJTYPE~NAME1~NAME2~ATTRTYPE is used as a key, d.h. OBJTYPE_PK ist nicht im Schluessel enthalten

- **tcLines - Example -** = { 'KNOT~NAME1~~PH': {'label': 'VL', 'color': 'red', 'linestyle': '-', 'linewidth': 3}

}

Definition der y-Achsentypen (y-Axes):

- werden ermittelt aus den verschiedenen ATTRTYPES in tcLines
- ATTRTYPE - z.B. 'PH' - wird dabei als Bezeichner für den Achsentyp benutzt
- die Achsen werden erstellt in der Reihenfolge in der sie in tcLines auftreten
- yTwinedAxesPosDeltaHPStart: (i.d.R. negativer) Abstand der 1. y-Achse von der Zeichenfläche; default: -0.0125
- yTwinedAxesPosDeltaHP: (i.d.R. negativer) zus. Abstand jeder weiteren y-Achse von der Zeichenfläche; default: -0.05

Attribute:

- alle gültigen

- –
- forceYType
- offset
- factor
- timeStart
- timeEnd
- legendInfosFmt
- label

AXES: pAx: Axes to be plotted on; if not specified: gca() is used

x-Achsen-Formatierung:

majLocator - Beispiele: mdates.MinuteLocator(interval=5)
mdates.MinuteLocator(byminute=[0,5,10,15,20,25,30,35,40,45,50,55])

majFormatter - Beispiele: mdates.DateFormatter('%d.%m.%y: %H:%M')

xTicksLabelsOff: wenn True, dann keine x-Achsen TickLabels

Return: yAxes: dct with AXES; key=y-Achsentypen yLines: dct with Line2Ds; key=Index from tcLines
vLines: dct with Line2Ds; key=Index from vLines yLinesLegendLabels: dct with Legendlabels;
key=Index from tcLines

```
>>> # -q -m 0 -s pltTC -y no -z no -w DHNetwork
>>> import pandas as pd
>>> import matplotlib
>>> import matplotlib.pyplot as plt
>>> import matplotlib.gridspec as gridspec
>>> import matplotlib.dates as mdates
>>> import math
>>> try:
...     import Rm
... except ImportError:
...     from PT3S import Rm
>>> # ---
>>> # xm=xms['DHNetwork']
>>> mx=mxs['DHNetwork']
>>> sir3sID=mx.getSir3sIDFromSir3sIDoPK('ALLG~~~LINEPACKGEOM') # 'ALLG~~~
↪5151766074450398225~LINEPACKGEOM'
>>> # mx.df[sir3sID].describe()
>>> # mx.df[sir3sID].iloc[0]
>>> plt.close()
>>> fig=plt.figure(figsize=Rm.DINA3q,dpi=Rm.dpiSize)
>>> gs = gridspec.GridSpec(3, 1)
>>> # -----
>>> axTC = fig.add_subplot(gs[0])
>>> yAxes,yLines,vLines,yLinesLegendLabels=Rm.Rm.pltTC(mx.df
... ,tcLines={
...     'ALLG~~~LINEPACKRATE':{'label':'Linepackrate','color':'red' ,
↪'linestyle':'-','linewidth':3,'drawstyle':'steps','factor':10}
...     , 'ALLG~~~LINEPACKGEOM':{'label':'Linepackgeometrie','color':'b' ,
↪'linestyle':'-','linewidth':3,'offset':-mx.df[sir3sID].iloc[0]
...     , 'timeStart':mx.df.index[0]+pd.Timedelta('10 Minutes')
...     , 'timeEnd':mx.df.index[-1]-pd.Timedelta('10 Minutes')}
...     , 'RSLW~wNA~~XA':{'label':'RSLW~wNA~~XA','color':'lime','forceYType
↪': 'N'}
```

(continues on next page)

(continued from previous page)

```

...     , 'PUMP~R-A-SS~R-A-DS~N': {'label': 'PUMP~R-A-SS~R-A-DS~N', 'color':
→ 'aquamarine', 'linestyle': '--', 'legendInfosFmt': '{:4.0f}'}
... }
... , pAx=axTC
... , vLines={
...     'a vLine Label': {'time': mx.df.index[0] + pd.Timedelta('10 Minutes
→')}
...     , 'color': 'dimgrey'
...     , 'linestyle': '--'
...     , 'linewidth': 5.}
... }
... , majLocator=mdates.MinuteLocator(byminute=[0,5,10,15,20,25,30,35,40,
→45,50,55])
... , majFormatter=mdates.DateFormatter('%d.%m.%y: %H:%M')
... #, xTicksLabelsOff=True
... )
>>> sorted(yAxes.keys())
['LINEPACKGEOM', 'LINEPACKRATE', 'N']
>>> sorted(yLines.keys())
['ALLG~~~LINEPACKGEOM', 'ALLG~~~LINEPACKRATE', 'PUMP~R-A-SS~R-A-DS~N',
→ 'RSLW~wNA~~XA']
>>> sorted(vLines.keys())
['a vLine Label']
>>> gs.tight_layout(fig)
>>> plt.show()
>>>

```

pltNetDHUS (kws)**

Plot: Net: DistrictHeatingUnderSupply.

Args (optional):

TIMES (als TIMEDELTA zu Szenariumbeginn):

- timeDeltaToRef: Reference Scenariotime (for MeasureInRefPerc-Calculations) (default: pd.to_timedelta('0 seconds'))
- timeDeltaToT: Scenariotime (default: pd.to_timedelta('0 seconds'))

FWVB

- **pFWVBFilterFunction: Filterfunction to be applied to FWVB to determine the FWVB to be plotted**

– default: `lambda df: (df.CONT_ID.astype(int).isin([1001])) & (df.WOLFK>0)`

* **CONT_IDisIn: [1001]**

- um zu vermeiden, dass FWVB aus Bloecken gezeichnet werden (unwahrscheinlich, dass es solche gibt)

* **WOLFK>0:**

- um zu vermeiden, dass versucht wird, FWVB mit der Soll-Leistung 0 zu zeichnen (pFWVBAttribute default is 'WOLFK')

FWVB Attribute (Size, z-Order)

- pFWVBAttribute: columnName (default: 'WOLFK')

- the column must be able to be converted to a float
- the conversion is done before FilterFunction
- see **ApplyFunction** and **NaNValue** for conversion details:

- * **pFWVBAttributeApplyFunction:** Function to be applied to column pFWVBAttribute

- default: lambda x: pd.to_numeric(x,errors='coerce')

- * **pFWVBAttributeApplyFunctionNaNValue:** Value for NaN-Values produced by pFWVB

- default: 0

- `.fillna(pFWVBAttributeApplyFunctionNaNValue).astype(float)`
is called after `ApplyFunction`

- pFWVBAttributeAsc: z-Order (default: False d.h. “kleine auf große”)
- pFWVBAttributeRefSize: scatter Sy-Area in pts^2 of for RefSizeValue (default: 10**2)
 - corresponding RefSizeValue is `Attribute.std()` or `Attribute.mean()` if `Attribute.std()` is < 1

FWVB (plot only large (small, medium) FWVB ...)

- `quantil_pFWVBAttributeHigh <=` (default: 1.)
- `quantil_pFWVBAttributeLow >=` (default: .0)
- default: all FWVB are plotted
- note that `Attribute > 0` is a precondition

FWVB Measure (Color)

- pFWVBMeasure (default: 'FWVB~*~*~*~W')
 - `float()` must be possible
- pFWVBMeasureInRefPerc (default: True d.h. Measure wird verarbeitet in Prozent T zu Ref)
 - 0-1
 - if refValue is 0 than refPerc-Result is set to 1
- pFWVBMeasureAlpha/Colormap/Clip
- 3Classes
 - **pFWVBMeasure3Classes (default: False)**
 - * **False:**
 - Measure wird nicht in 3 Klassen dargestellt
 - die Belegung von `MCategory` gemaess `FixedLimitsHigh/Low` erfolgt dennoch
 - CatTexts (werden verwendet wenn 3Classes Wahr gesetzt ist)
 - * für `CBLegend (3Classes)` als `_zusätzliche` Beschriftung rechts
 - * als Texte für die Spalte `MCategory` in return pFWVB

- * pMCatTopText
- * pMCatMidText
- * pMCatBotText
- CatAttribs (werden verwendet wenn 3Classes Wahr gesetzt ist)
 - * für die Knotendarstellung
 - * pMCatTopAlpha/Color/Clip
 - * pMCatMidAlpha/Colormap/Clip
 - * pMCatBotAlpha/Color/Clip
- CBFixedLimits
 - pFWVBMeasureCBFixedLimits (default: False d.h. Farbskala nach vorh. min./max. Wert)
 - * wird Wahr gesetzt sein, wenn 3Classes Wahr gesetzt ist
 - * damit die mittlere Farbskala den Klassengrenzen “gehört”
 - pFWVBMeasureCBFixedLimitLow (default: .10)
 - pFWVBMeasureCBFixedLimitHigh (default: .95)

CB

- CBFraction: fraction of original axes to use for colorbar (default: 0.05)
- CBHpad: fraction of original axes between colorbar and new image axes (default: 0.0275)
- CBLLabelPad (default: -50)
- CBTicklabelsHPad (default: 0.)
- CBAAspect: ratio of long to short dimension (default: 10.)
- CBSShrink: fraction by which to shrink the colorbar (default: .3)
- CBAAnchorHorizontal: horizontal Fußpunkt der colorbar in Plot-% (default: 0.)
- CBAAnchorVertical: vertikaler Fußpunkt der colorbar in Plot-% (default: 0.2)

CBLegend (3Classes) - Parameterization of the representative Symbols

- CBLe3cTopVPad (default: $1+1*1/4$)
- CBLe3cMidVPad (default: .5)
- CBLe3cBotVPad (default: $0-1*1/4$)
 - 1 is the height of the Colorbar
 - the VPads (the vertical Sy-Positions) are defined in cax.transAxes Coordinates
 - cax is the Colorbar Axes
- CBLe3cSySize= $10**2$ (Sy-Area in pts^2)
- CBLe3cSyType='o'

ROHR

- **pROHRFilterFunction:** Filterfunction to be applied to PIPEs to determine the PIPEs to be plotted
 - **default:** `lambda df: (df.KVR.astype(int).isin([2])) & (df.CONT_ID.astype(int).isin([1001]))`
 - * **KVRisIn: [2]**
 - 1: supply-line
 - 2: return-line
 - * **CONT_IDisIn: [1001]**
 - um zu vermeiden, dass Rohre aus Bloecken gezeichnet werden (deren Koordinaten nicht zu den Koordinaten von Rohren aus dem Ansichtsblock passen)
 - * **DI>0:**
 - um zu vermeiden, dass versucht wird, Rohre mit dem Innendurchmesser 0 zu zeichnen (pROHRAttribute default is 'DI')

ROHR (PIPE-Line: Size and Color, z-Order)

- **pROHRAttribute: columnName (default: 'DI')**
 - the column must be able to be converted to a float
 - the conversion is done before FilterFunction
 - see **ApplyFunction** and **NaNValue** for conversion details:
 - * **pROHRAttributeApplyFunction:** Function to be applied to column pROHRAttribute
 - default: `lambda x: pd.to_numeric(x,errors='coerce')`
 - * **pROHRAttributeApplyFunctionNaNValue:** Value for NaN-Values produced by pROHRAttributeApplyFunction
 - default: 0
 - `.fillna(pROHRAttributeApplyFunktionNaNValue).astype(float)` is called after ApplyFunction
- **pROHRAttributeAsc:** z-Order (default: False d.h. "kleine auf grosse")
- **pROHRAttributeLs** (default: '-')
- **pROHRAttributeRefSize:** plot linewidth in pts for RefSizeValue (default: 1.0)
 - corresponding RefSizeValue is `Attribute.std()` or `Attribute.mean()` if `Attribute.std()` is < 1
- **pROHRAttributeColorMap** (default: `plt.cm.binary`)
- **pROHRAttributeColorMapUsageStart** (default: 1./3; Wertebereich: [0,1])
 - Farbskala nach vorh. min./max. Wert
 - die Farbskala wird nur ab UsageStart genutzt
 - d.h. Werte die eine "kleinere" Farbe haetten, bekommen die Farbe von UsageStart

ROHR (plot only large (small, medium) pipes ...)

- quantil_pROHRAttributeHigh <= (default: 1.)
- quantil_pROHRAttributeLow >= (default: .75)
- default: only the largest 25% are plotted
- note that Attribute >0 is a precondition

ROHR (PIPE-Marker: Size and Color)

- pROHRMeasure columnName (default: 'ROHR~*~*~*~QMAV')
- pROHRMeasureApplyFunction: Function to be applied to column pROHRMeasure (default: lambda x: math.fabs(x))
- pROHRMeasureMarker (default: '.')
- pROHRMeasureRefSize: plot markersize for RefSizeValue in pts (default: 1.0)
 - corresponding RefSizeValue is Measure.std() or Measure.mean() if Measure.std() is < 1
 - if pROHRMeasureRefSize is None: plot markersize will be plot linewidth
- pROHRMeasureColorMap (default: plt.cm.cool)
- pROHRMeasureColorMapUsageStart (default: 0.; Wertebereich: [0,1])
 - Farbskala nach vorh. min./max. Wert
 - die Farbskala wird nur ab UsageStart genutzt
 - d.h. Werte die eine "kleinere" Farbe hätten, bekommen die Farbe von UsageStart

NRCVs - NumeRiCal Values to be displayed

- **pFIGNrcv: List of Sir3sID RegExps to be displayed (i.e. ['KNOT~PKON-Knoten~S*~S+~QM'])** d the 1st Match is used if a RegExp matches more than 1 Channel
further Examples for RegExps (and corresponding Texts):
 - WBLZ~WärmeblnzGes~S*~S+~WES (Generation)
 - WBLZ~WärmeblnzGes~S*~S+~WVB (Load)
 - WBLZ~WärmeblnzGes~S*~S+~WVERL (Loss)

WBLZ~[S]+~S*~S+~S+: Example for a RegExp matching all Channels with OBJTYPE WBLZ
- pFIGNrcvTxt: corresponding (same length required!) List of Texts (i.e. ['Kontrolle DH']) default: None
- **pFIGNrcvFmt (i.e. '{:12s}: {:8.2f} {:6s}')**
 - Text (from pFIGNrcvTxt)
 - Value
 - UNIT (determined from Channel-Data)
- **pFIGNrcvPercFmt (i.e. '{:6.1f}%')**
 - ValueInRefPercent
 - if refValue==0: 100%
- pFIGNrcvXStart (.5 default)

- pFIGNrcvYStart (.5 default)

Category - User Heat Balances to be displayed

- pFWVBGCategory: List of Heat Balances to be displayed (i.e. ['BLNZ1u5u7'])
default: None
- pFWVBGCategoryUnit: Unit of all these Balances (default: ['kW'])
- pFWVBGCategoryXStart (.1 default)
- pFWVBGCategoryYStart (.9 default)
- pFWVBGCategoryCatFmt (i.e. '{:12s}: {:6.1f} {:4s}')
- Category NAME
- Category Load
- pFWVBGCategoryUnit
- pFWVBGCategoryPercFmt (i.e. '{:6.1f}%')
- Last Ist/Soll
- pFWVBGCategory3cFmt (i.e. '{:5d}/{:5d}/{:5d}')
- NOFTops
- NOFMids
- NOFBots

VICs - VeryImportantCustomers whose Values to be displayed

- pVICsDf: DataFrame with VeryImportantCustomers (Text & Specification)

columns expected:

- Kundename (i.e. 'VIC1') - Text
- Knotenname (i.e. 'V-K007') - Specification by Supply-Node

i.e.: pd.DataFrame({'Kundename': ['VIC1'], 'Knotenname': ['V-K007']})

- pVICsPercFmt (i.e. '{:12s}: {:6.1f}%')
- Kundename
- Load in Percent to Reference
- pVICsFmt (i.e. '{:12s}: {:6.1f} {:6s}')
- Kundename
- Load
- pFWVBGCategoryUnit
- pVICsXStart (.5 default)
- pVICsYStart (.1 default)

Figure:

- pltTitle: title [not supitle] (default: 'pltNetFigAx')
- figFrameon: figure frame (background): displayed or invisible (default: True)

- `figEdgecolor`: edge color of the Figure rectangle (default: 'black')
- `figFacecolor`: face color of the Figure rectangle (default: 'white')

Returns:**pFWVB**

- **columns changed (compared to vFWVB):**

- `pFWVBAttribute` (wg. z.B. `pFWVBAttributeApplyFunction` und `.astype(float)`)

- **columns added (compared to vFWVB):**

- `Measure` (in % zu Ref wenn `pFWVBMeasureInRefPer=True`)
- `MeasureRef` (Wert von `Measure` im Referenzzustand)
- `MeasureOrig` (Wert von `Measure`)
- **MCategory: str (Kategorisierung von Measure mit FixedLimitHigh/Low-Werten):**

- * `TopText` or

- * `MidText` or

- * `BotText`

- `GCategory`: list (non-empty only if req. `GCategories` are a subset of the available `Categories` and object belongs to a req. `Category`)
- `VIC` (filled with `Kundenname` from `pVICsDf`)

- **rows (compared to vFWVB):**

- `pFWVB` enthaelt dieselben Objekte wie `vFWVB`
- aber: die geplotteten Objekte sind ggf. nur eine Teilmenge (wg. z.B. `pFWVBFilterFunction`)

exception `Rm.RmError` (*value*)

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

m

Mx, ??

r

Rm, ??

x

Xm, ??