

A series of horizontal bars of varying lengths and colors (teal, blue, and dark blue) are positioned on the left side of the slide, creating a modern, abstract background element.

# Reading XML & SDMX Data with SAS

Paul Van Mol



# SDMX – Statistical Data and Metadata eXchange

[Sdmx.org](http://Sdmx.org)

- ISO Standard to describe statistical data and metadata
- ISTAT offers a SDMX helper and %gettimeseries() macro to retrieve SDMX data from different providers
- SAS Econometrics provides libname several engines
- SDMX-ML allows retrieval of SDMX timeseries in XML and JSON format



# Reading SDMX Data

with SDMX Helper and %gettimeseries()

- <https://github.com/amattioc/SDMX>
- SDMX Helper: select Data Provider
- Data Flow and Dimensions of the Time Series like currency=USD
- Actions Menu: Build Commands
- The command in SAS uses the %gettimeseries () macro call:

The screenshot shows the SDMX Helper Tool interface. The 'Providers' tab is active, and the 'Query' section shows 'Provider: ECB' and 'EXR/USD...'. The 'Dataflow selection' section has a filter 'exchange rate' and a table of dataflows. The 'Dimension selection' section shows a table of dimensions with '1 CURRENCY' selected. The 'Code list selection' section shows a table of codes with 'USD' selected. The bottom status bar shows the query URL and the time of the query.

SDMX Helper Tool

Providers Actions Help

Query

Provider: ECB EXR/USD...

Dataflow selection

Filter flows by name or description: exchange rate

Dataflow	Version	Agency	Description
EXR	1.0	ECB	Exchange Rates
EXR_PUB	1.0	ECB.DISS	Exchange Rates - Pub
EXR_PUB	1.0	ECB.DISS	Exchange Rates - Pub
MOBILE_EXR	1.0	ECB.DISS	Exchange rates
EXR	1.0	ECB	Exchange Rates

Dimension selection

Dimension to edit: Clear selected dimension

Dimension	Description
0/FREQ	Frequency
1/CURRENCY	Currency
2/CURRENCY_DENOM	Currency denominator
3/EXR_TYPE	Exchange rate type
4/EXR_SUFFIX	Series variation - EXR context

Code list selection

Filter codes (1/14): usd

Code ID	Code Description
<input checked="" type="checkbox"/> USD	US dollar
<input type="checkbox"/> X2	All currencies except USD
<input type="checkbox"/> X4	All currencies except EUR, USD
<input type="checkbox"/> X5	All currencies except EUR, JPY, USD
<input type="checkbox"/> X6	All currencies except EUR, CHF, GBP, JPY, USD
<input type="checkbox"/> X7	All currencies except EUR, USD, JPY, GBP, CHF, domes
<input type="checkbox"/> X8	All currencies except USD, EUR, GBP, JPY, CHF, domes

Jun 05, 2022 9:57:50 PM it.bancaditalia.oss.sdmx.client.RestSdmxClient runQuery  
INFO: Contacting web service with query: https://sdw-wsrest.ecb.europa.eu/service/datastructure/ECB/ECB\_EXR/1.0?references=children

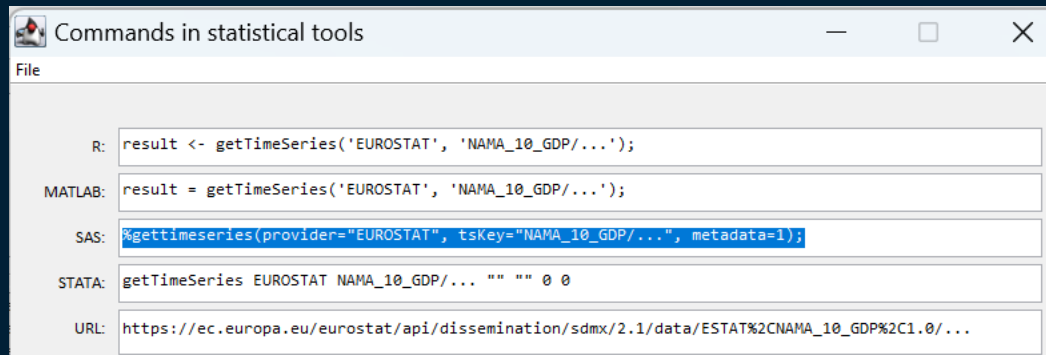
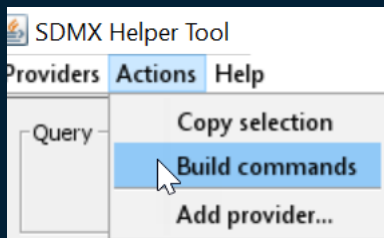
# Reading SDMX Data

with SDMX Helper and %gettimeseries()

- <https://github.com/amattioc/SDMX>

Actions Menu: Build Commands

The command in SAS uses the  
%gettimeseries () macro call:



# Reading SDMX Data

## with SDMX Helper and %gettimeseries()

```
/*create the SDMX folder in the home directory of the user*/
%let homedir=%sysget(HOME);
%let path=&homedir;
options dlcreatedir;
libname data "&path/SDMX";

/*clone the SDMX git repository to a local folder*/
data _null_;
RC = GITFN_CLONE("https://github.com/amattioc/SDMX.git",
"&path/SDMX");
run;

/* example Getting Exchange Rates from ECB data provider*/
%gettimeseries(provider="ECB", tsKey="EXR.A.USD.EUR.SP00.A",
metadata=1);
```

# Reading SDMX Data

using SAS ETS Data Access engines

- IMF data (14742 series) & ECB (13 series) can be read using SASEFRED engine
  - FRED website here: "<https://fred.stlouisfed.org/tags/series?t=imf>"
  - [The SASEFRED Interface Engine](#)
- ECB data (13 series) can be read using SASEFRED engine- see FRED website here:  
"<https://fred.stlouisfed.org/tags/series?t=ecb>"
- ECB data (212,000 series) can be read using SASEQUAN engine.
  - Using the NASDAQ api , see the following documentation page:  
<https://data.nasdaq.com/data/ECB-european-central-bank/documentation>
- Retrieving OECD Data using SASEOECD engine
  - [https://documentation.sas.com/doc/en/etsug/15.2/etsug\\_saseoecd\\_examples01.htm](https://documentation.sas.com/doc/en/etsug/15.2/etsug_saseoecd_examples01.htm)

# Reading SDMX Data

using SASEFRED for IMF series

- IMF data (14742 series) & ECB (13 series) can be read using SASEFRED engine
  - FRED website here: "<https://fred.stlouisfed.org/tags/series?t=imf>"
  - [The SASEFRED Interface Engine](#)
- Create an account + create API key
- Select time series

# Reading SDMX Data

## Using SASEFRED libname engine

```
title 'Retrieve Balance of Payment Data for the Exports and Imports';
libname _all_ clear;
libname fred sasefred "&path\viyawhatsnew\sdmx_query\freddata"
OUTXML=fredex01 AUTOMAP=replace
MAPREF=MyMap XMLMAP= "&path\viyawhatsnew\sdmx_query\freddata\fredex01.map"
APIKEY='399eb04a24a59583574beea2248db31'
IDLIST='bopxgs,bopmgs'
START='1997-01-01' END='2011-01-01'
FREQ='a' OUTPUT=1 AGG='avg' FORMAT=xml;
data export_import;
set fred.fredex01 ;
run;
proc contents data=export_import; run;
proc print data=export_import; run;
```



# Reading SDMX Data

using SASEQUAN

- ECB data (212,000 series) can be read using SASEQUAN engine.
  - Using the NASDAQ api , see the following documentation page:  
<https://data.nasdaq.com/data/ECB-european-central-bank/documentation>
  - Create an account + create API key
- Data Organization
- [https://data.nasdaq.com/data/{Time-Series\\_Code}](https://data.nasdaq.com/data/{Time-Series_Code}).
- For a complete list of Time-Series Codes included in this data feed, use:
- [https://data.nasdaq.com/api/v3/databases/ECB/metadata?api\\_key=mSqQTWCPgzkiy2KxUjzM](https://data.nasdaq.com/api/v3/databases/ECB/metadata?api_key=mSqQTWCPgzkiy2KxUjzM)

# Reading SDMX Data

## Using SASEFRED libname engine

```
title 'Historical Prices for Oil India Limited';
libname _all_ clear;
libname mylib "C:/workshop/viyawhatsnew/sdmx_query/quant/doc";
libname myQoil sasequan "C:/workshop/viyawhatsnew/sdmx_query/quant/test"
    apikey='mSqqTwCPgzkiy2KxUjzM'
    idlist='NSE/OIL'
    format=XML outXml=oil
    automap=replace mapref=MyMap
    xmlmap="C:/workshop/viyawhatsnew/sdmx_query/quant/oil.map"
    start='2013-09-01'
    end='2013-11-05'
    freq='daily'
    collapse='daily'
;
data mylib.oilall;
    set myQoil.oil;
run;
```

# Reading SDMX Data

using SAS ETS Data Access engines

- Retrieving OECD Data using SASEOECD engine
  - [https://documentation.sas.com/doc/en/etsug/15.2/etsug\\_saseoecd\\_examples01.htm](https://documentation.sas.com/doc/en/etsug/15.2/etsug_saseoecd_examples01.htm)

# Reading SDMX Data

## Using SASEOECD libname engine

```
libname oecddata base "&path/sdmx_query/oecddata";
data keylist0;
    length key0 $8;
    key0='EA17'; output; /* country is euro area; 17 countries */
run;
data keylist1;
    length key1 $8;
    key1='B1_GA'; output; /* transaction is GDP; output approach */
run;
data keylist2;
    length key2 $2;
    key2='C'; output; /* measure is current prices */
run;
title 'Request GDP for EA_17 in Current Prices';
LIBNAME myLib saseoecd "&path/sdmx_query/oecddata"
    setid=SNA_TABLE1_SNA93
    inset0=keylist0 inset1=keylist1
    inset2=keylist2 out=gstart
    ;
```

```
data myGDP;
    set myLib.gstart ;
run;
proc print data=myGDP; run;
```

# Reading SDMX Data

SDMX-ML and SAS XMLV2 Engine

- Retrieving OECD Data using SASEOECD engine
  - [https://documentation.sas.com/doc/en/etsug/15.2/etsug\\_saseoecd\\_examples01.htm](https://documentation.sas.com/doc/en/etsug/15.2/etsug_saseoecd_examples01.htm)
- [OECD Statistics](#)

# Reading SDMX Data

Using OECD XML Rest Api and XMLV2 libname engine

```
%let homedir=%sysget (HOME);
%let path=&homedir/viyawhatsnew;
%let xmlpath = &path/sdmx_query/;

filename map "&xmlpath.map.txt";
filename resp "&xmlpath.resp.txt";
proc http

URL="https://stats.oecd.org/restsdmx/sdmx.ashx/GetData/QNA/AUS+AUT.GDP+B1_GE.CU
R+VOBARS.A.Q/all?startTime=2009-Q2&endTime=2011-Q4&format=compact_v2"

METHOD="GET"
OUT=resp;
run;quit;

libname resp XMLv2 automap=REPLACE xmlmap=map;
proc datasets;
copy out=WORK in=resp;
run;quit;
```

# Using XMLV2 Automap to automatically read XML

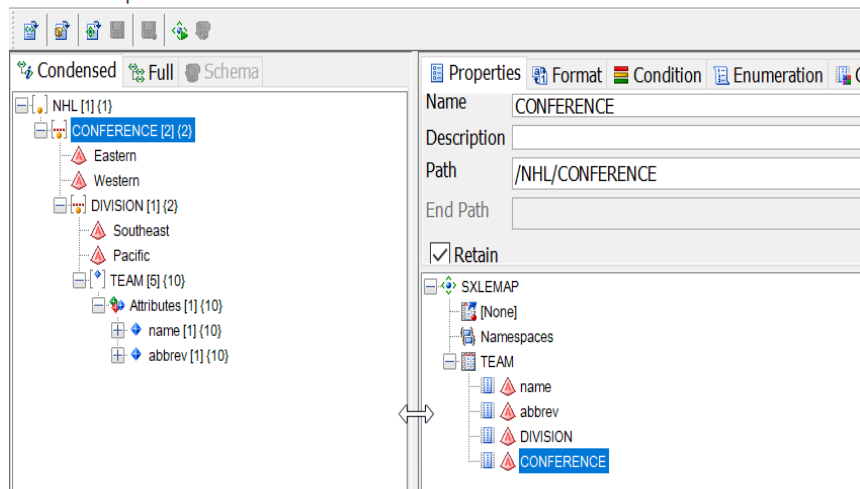
## 3 ways to Create XML MAPS

SAS XML Mapper	XMLV2 Automap=replace option	Manually Code XML Map using XPATH syntax
Based on XML File or XML Schema	Base on XML File (no XML Schema support)	
Automatic XML Map Creation	Automatic XML Map Creation	No Automatic all Manual
Drag and Drop Custom XML Map (single table)	No option to create single table	
Modify column Attributes	Modify column attributes manually Using XMLMAP XPATH Syntax	Modify column attributes manually Using XMLMAP XPATH Syntax

# Using XML Mapper

SAS XML Mapper

File Tools Help



```
/*REading xml as one sas dataset*/  
%let
```

```
path=c:\workshop\git\sasstudiogs;
```

```
filename nhl
```

```
"&path\xmlfiles\nhl.xml"; /*1*/
```

```
filename map
```

```
"&path\xmlfiles\nhl.map"; /*2*/
```

```
libname nhl xmlv2 xmlmap=map;
```

```
/*3*/
```

```
proc print data=nhl.teams; /*4*/
```

```
run;
```



# Using XMLV2 Automap

```
/*Using Automap to Generate an XMLMap*/
```

```
filename nhl "&path\xmlfiles\nhl.xml"; /**/  
filename map "&path\xmlfiles\nhlgenerate.map"; /**/  
libname nhl xmlv2 automap=replace xmlmap=map; /**/  
proc print data=nhl.team; run;  
proc print data=nhl.conference; run;  
/*Combine 3 tables */  
proc sql;  
create table work.allnhl as  
select * from nhl.division natural join nhl.conference  
natural join nhl.team;  
quit;
```

# Manually Modifying XML Maps

```
/*Step 3: Adjust the Entity CreationDate to Char*/  
/*Table Entity:  
<COLUMN name="EntityCreationDate">  
    <PATH  
syntax="XPathENR">/{2}LEIData/{2}LEIRecords/{2}LEIRecord/{2}Entity  
/{2}EntityCreationDate</PATH>  
    <TYPE>character</TYPE>  
    <DATATYPE>string</DATATYPE>  
    <LENGTH>29</LENGTH>  
    <FORMAT width="29">$CHAR</FORMAT>  
    <INFORMAT width="29">$CHAR</INFORMAT>  
</COLUMN>
```

# Tips and Tricks

- Run SDMXHelper from ISTAT:

SAS has builtin SASPrivateJavaRuntimeEnvironment:

following command:

- `"C:\Program Files\SASHome\SASPrivateJavaRuntimeEnvironment\9.4\jre\bin\java" -  
classpath c:\workshop\viyawhatsnew\sdmx_query\SDMX.jar  
it.bancaditalia.oss.sdmx.helper.SDMXHelper`

# Tips and Tricks

For the XMLV2 Engine Automap option, you need to allow the SAS JRE to allocate more memory: -xmx2048m instead of the default 128m

/\* Options used when SAS is accessing a JVM for JNI processing \*/

```
-JREOPTIONS=(  
    -Xms512m  
    -Xmx2048m  
)
```

A series of horizontal bars of varying lengths and colors (teal, blue, and dark blue) are arranged vertically on the left side of the slide, creating a decorative, abstract pattern.

[sas.com](https://sas.com)