SDMX Standards: Section 3A PaRT VII

SDMX-ML:

Schema and Documentation

Samples

Version 2.1

April 2011

© SDMX 2011

http://www.sdmx.org/

Contents

[1 Introduction 1](#_Toc290377119)

[2 Structure Samples 1](#_Toc290377120)

[3 Data Samples 2](#_Toc290377121)

[4 Metadata Sample 3](#_Toc290377122)

[5 Query Samples 3](#_Toc290377123)

# Introduction

This document provides a brief overview of the currently available set of sample files. It is intended that more samples will be added over time, and the explanations in this document updated. Eventually, this entire document will be migrated to a complete user guide which details the various aspects of SDMX, focusing on specific use cases and beginning to end uses of the standard.

# Structure Samples

The structure samples that are available serve to describe the structure of the data and reference metadata samples.

The directory common contains structural metadata based on the content oriented guidelines, and this is used across all samples.

For all samples, there is a common sample file which contains a subset of standard SDMX components. Tese are contained in common/common.xml. Other files may make external references to this file.

The directory exr contains 3 similar data structure definitions, each with very slight modifications. These data structures are all based on the ECB exchange rates and serve to demonstrate that the newly introduced attribute relationship construct. A summary of the files is as follows:

* exr/common/exr\_common.xml: this contains the structural metadata that is used across the various data structures. Each of the subsequent structure message provide external references to this file
* exr/ecb\_exr\_ng/ecb\_exr\_ng.xml: this contains an exchange rate data structure with no groups defined. All attribute relationships are to specific dimensions. The file ecb\_exr\_ng\_full.xml represents the same structure, but does not use external referencing to make the file easier to load into tools which cannot resolve local file references
* exr/ecb\_exr\_sg/ ecb\_exr\_sg.xml: this contains an exchange rate data structure with a sibling group defined. In this sample, only on attribute references the sibling group, and the rest maintain the relationships defined in the data structure without groups. The file ecb\_exr\_sg\_full.xml represents the same structure, but does not use external referencing to make the file easier to load into tools which cannot resolve local file references
* exr/ecb\_exr\_rg/ ecb\_exr\_rg.xml: this contains an exchange rate data structure with an addition group defined. In this sample only one attribute specifies a relationship with the new group, but the dimensions from the previous sample now declare the sibling group as an attachment group. The file ecb\_exr\_rg\_full.xml represents the same structure, but does not use external referencing to make the file easier to load into tools which cannot resolve local file references

The directory demography contains a data structure and metadata structure. The summary of the structural metadata files are as follows:

* demography.xml: this contains the demography data structure definition. This data structure definition declares measure dimension. It is also worth noting that the measure concepts have different core representations. This is important when the sample data is discussed.
* esms.xml: this contains a much simplified variation of the Eurostat SDMX Metadata Structure. It feature all of the new features of the metadata structure including the various target objects, the ability to reference multiple possible targets from a report structure and the ability to have more complex structures with repeating attributes, attributes with both sub content and values, and attributes with structured text as it content

# Data Samples

The exchange rate sample data files all contains the same set of data, and each variation of the key family has 8 corresponding sample files intended to demonstrate the effects of the various orientations. The difference in the data structures is discussed in the previous section. For any given data structure, the following samples files exist (note the names vary based on the data structure used and the format, but follow the same structure (note [format] is generic or structured and [dsd] is ecb\_exr\_ng, ecb\_exr\_sg, or ecb\_exr\_rg.:

* [format]/[dsd]\_flat.xml: this represents ungrouped observations, where every observation contains a full key. It is worth noting how the use of groups changes change the verbosity of the message
* [format]/[ [dsd]\_ts.xml: this represents the data organised as time series, using the time series only message
* [format]/[dsd]\_ts.xml: this represents the same orientation as the previous file, but uses the general format. It is worth noting that this file is exactly that same as the previous file with the exception of the root element
* [format]/[ [dsd]\_xs.xml: this file shows the data represented with another dimension at the observation level. Because of the attribute relationships, this results in the duplication of values. But as the data structures introduce groups, this impact is reduced

Note that for all structured data, the data structure specific schemas have same file name, but with a .xsd extension.

The demography data has only on instance associated with it, and that is a cross sectional format with explicit measures being used. It is important to examine the data structure definition schema (demography\_xs\_ex.xsd) to note how the various measure have different representations for the observed value (OBS\_VALUE). By modifying the sample, you can see where this provides extra validation.

# Metadata Sample

There is a single set of reference metadata samples based on the ESMS metadata structure in the demography folder. This is meant to represent a very simplified metadata report that is attached to the demography data via a category in which the data flow is defined. These files are:

* esms\_generic.xml: this the reference metadata in the generic format
* esms\_structured.xml: this is the same data in the metadata structure specific format (note that esms.xsd is the schema for this)

By comparing the files, you can see the similarities between the two formats. In fact, the only difference you will notice (outside of the namespace) is that the metadata structure specific format has element names based on the metadata attribute identifiers where as the generic structure has this information in a ReportedAttribute element.

An examination of the metadata structure specific schema will help in understanding the rules for creating these schemas, as the schema is annotated with comments explaining the various features. It is also worth noticing how the representation scheme for the target object which references a category resulted in a reference which only allows the categories from that scheme to be referenced.

Note that the xhtml folder contains the xhtml schemas so that the structure text content can be validated.

# Query Samples

The query directory contains a set of query and response documents that highlight some of the new features in the query. Each query is paired with it response by the file name (e.g. query\_esms.xml is the query and response.esms.xml is the response). The purpose of the queries is as follows:

* query\_demo\_stub.xml: this is a simple query which demonstrates how one can check for the existence of an object by simply requesting that no references be resolved and only the stub be returned. The intention of this query is to simply find out what is the version of the currently active demography data structure.
* query\_esms\_shallow.xml: this is a query for the ESMS metadata structure, which is returned in full detail. Only the object referenced directly from the metadata structure are requested and returned, and only the stubs of the objects.
* query\_esms\_deep.xml: this is an example of a where used query. The intention is to query for any objects which are used directly or indirectly by the ESMS metadata structure. You will notice that the metadata structure is requested to not be returned. Instead, only the stubs of all objects referenced form the metadata structure and the object which they reference (and so on) are returned.
* query\_cl\_all.xml: this a query for a subset of the very large area codelist. Notices that the return details specify that the matched item should be cascaded down the hierarchy, meaning all of its child codes should be returned. In this example, this results in all regions and sub-regions for Greece.
* query\_cl\_regions.xml: this is query for only the regions that are direct children of Greece. The query uses the parent property of the code to find the administrative regions within the country. Notice that the results are not cascaded, so only one level is returned.