



Statistical Working Group / Technical Working Group

POSSIBLE WAYS OF IMPLEMENTING CL_OBS_STATUS CODE LIST

20 OCTOBER 2014

1) Introduction

First of all, it is important to note that the "Observation status" code list has an heterogeneous character as it mixes concepts which are not always mutually exclusive (e.g. a missing value can generate a break in time series, an estimated value can be of low reliability). Thus, to cope with the issue of allocating more than one flag to one statistical value, this code list should ideally be broken down into various sub-code lists corresponding to the various concepts covered. It was not done so because it was felt that it would unnecessarily increase the number of (very short) code lists for low benefits in terms of technical and conceptual orthodoxy.

However, in view of the central importance of this code list, it is essential to provide implementers with all possible ways of implementing this code list so that they can decide, based on their specific implementation needs, which option best suits their requirements. These various options are presented in the sections below, and their pros and contras explicated.

In case implementers are satisfied with one flag per observation value, they are invited to apply the recommended hierarchy proposed under "2) One flag only per value". In the case of multiple flagging, although the three options described below are in theory applicable, they should certainly not be considered equally; indeed, option 3.1 "Duplication approach" is to be considered as the recommended general solution; in cases where implementers do not think the recommended general solution is able to be applied, or appropriate to apply, in their particular context, an acceptable alternative solution called "Decomposition approach" is proposed. The third option, called "extended single code list approach", is documented here for the sake of completeness but strongly discouraged.

The SDMX standard allows for the use of zero or more observation level attributes, using any identifiers. However, SDMX-EDI imposes the mandatory use of the observation level attribute called OBS_STATUS. In the past, SDMX-EDI has limited itself, for practical reasons, to the use of the observation level attributes OBS_STATUS, CONF_STATUS, PRE_BREAK_VALUE and COMMENT_OBS, but SDMX-EDI can handle any number of observation level attributes, as long as OBS_STATUS is included.

For backward compatibility between SDMX-ML and SDMX-EDI, the observation level attribute OBS_STATUS must be included in every Data Structure Definition (DSD) as mandatory. Without it, backward compatibility is not possible.

2) One flag only per value

In case implementers want to use only one single flag per value, they should use the hierarchy below to determine the code to be used. This approach (choice of only one event, namely the most important one) offers a good compromise between simplicity for the user, completeness of provided information and presentational easiness of management on the user interface side. The main drawback of this approach is the loss of information resulting from the use of only one flag when several flags may apply to a given value.

Example: From now on, value x is compiled on the basis of a methodology diverging from the previous one (e.g. following an alignment with international standards), which generates a break in time series. In this case, two flags, namely B (*Time series break*) and D (*Definition differs*), should be used. If only one flag is to be indicated, then use should be made of the hierarchy below to determine which flag to use. In this case, this would be B since B has precedence over D in the hierarchy.

Observation status hierarchy	Relevant in conjunction with...	
	numeric values	missing values
B / time series break (highest importance)	Yes	Yes
O / missing value		Yes
M / missing value; data cannot exist		Yes
L / missing value; data exist but were not collected		Yes
H / missing value; holiday or weekend		Yes
Q / missing value; suppressed		Yes
J / derogation	Yes	Yes
S / strike and other special events	Yes	Yes
D / definition differs	Yes	
I / imputed value	Yes	
F / forecast value	Yes	
E / estimated value	Yes	
P / provisional value	Yes	
N / not significant	Yes	
U / low reliability	Yes	
V / unvalidated value	Yes	
G / experimental value	Yes	
A / normal value	Yes	

3) Multiple flagging

There might be cases however where implementers will want to attach multiple flags to one statistical value. To cope with this situation, three solutions have been analysed, based on :

- (1) a duplication approach;
- (2) a decomposition approach;
- (3) an extended single code list approach.

Technically the three approaches are possible. However, considering the severe limitations that the third approach would implicate, only one of the first two approaches will be recommended for use (as said earlier also with a view to improving harmonisation across implementations).

3.1) Duplication approach (recommended solution)

In this case, the OBS_STATUS concept is duplicated as many times as needed. These duplicated concepts can be named "OBS_STATUS_1", "OBS_STATUS_2", "OBS_STATUS_3", etc. All these concepts have to be inserted in the DSD and linked to the CL_OBS_STATUS code list. Only one value is allowed per code list.

The main advantages of this solution are its simplicity and the fact that it does not require listing the possible combinations.

Drawbacks are the multiplication of the same concept and the absence of implicit checks which makes it possible to enter aberrant combinations of codes (e.g. normal value and low reliability).

This approach is the recommended general solution for implementations where multiple flagging is required.

3.2) Decomposition approach (accepted but not preferred solution)

Here, CL_OBS_STATUS code list is broken down into its basic components, distinguished on the basis of the different concepts used and their mutually exclusive character. The list of "building blocks" composing the CL_OBS_STATUS code list as it stands at present could be represented as separate concepts as follows:

- Concept OBS_STATUS (Observation status) → code list CL_OBS_MAIN (A,E,G,H,I,J,M,O, L,Q,S): these codes can be grouped in one single code list because they are mutually exclusive: a normal value cannot be estimated nor imputed nor missing; an estimated value cannot be normal nor imputed nor missing, an imputed value cannot be normal nor estimated nor missing, etc.

For the other status codes, a single Boolean code list¹ can be created to enable / disable a specific flag:

- Concept OBS_BREAK → code list CL_BOOLEAN, with code Y corresponding to flag B (Time series break);
- Concept OBS_DEF_DIFFERS → code list CL_BOOLEAN, with code Y corresponding to flag D (Definition differs);

¹ CL_BOOLEAN with 2 codes: Y (Yes), N (No)

- Concept OBS_FORECAST → code list CL_BOOLEAN, with code Y corresponding to flag°F (Forecast value);
- Concept OBS_PROV → code list CL_BOOLEAN, with code Y corresponding to flag P (Provisional value);
- Concept OBS_SIGNIFICANCE → code list CL_BOOLEAN, with code Y corresponding to flag N (Not significant);
- Concept OBS_VALIDATION→ code list CL_BOOLEAN, with code Y corresponding to flag V (Unvalidated value);
- Concept OBS_RELIABILITY → code list CL_BOOLEAN, with code Y corresponding to flag U (Low reliability).

If additional flags are needed, more concepts can be defined accordingly. All these concepts have to be inserted in the DSD and linked to CL_BOOLEAN.

The main advantage of this proposal is its full compliance with the technical standards and the content-oriented guidelines which insist on separating concepts which are different in content. Drawbacks are the multiplication of (very) small code lists and the absence of implicit checks which makes it possible to enter aberrant combinations of codes (e.g. normal value **and** low reliability). Furthermore, any new code will require reconsidering the content of the various sub-code lists.

Although not recommended as the preferred solution, this approach can be implemented in cases where the general solution cannot be applied, or is not the appropriate solution, in a particular context.

Comments on the choice of the recommended solution

Both "Decomposition" and "Duplication" options provide acceptable workarounds to the problem of multiple flagging, and appear to be quite similar in practice. The trade-off in this context was between orthodoxy and ease of implementation.

Conceptually the "Decomposition" approach is definitely the strongest of the two as it not only allows separating concepts, but also helps arranging codes into more homogeneous code lists. It also requires that implementers define pure concepts and name them accordingly.

This document recommends the "Duplication" approach mainly on the practical grounds of ease of implementation in the current state of development of the technical standard. This means that the recommended approach could be reconsidered in the future, would the technical standard better accommodate the decomposed approach.

3.3) Extended single code list approach (strongly discouraged)

The extended version of CL_OBS_STATUS (see below) provides the full list of logically possible combinations of codes in a specific SDMX implementation.

An advantage of this solution would be that only meaningful combinations of flags are included in the list. Users would not be able to choose combinations which would not make sense (such as "missing" and "estimated").

However, there are several drawbacks related to the technical implementation of this solution:

- relative complexity for users to find the right combination of flags
- maintenance burden in case of revision of the code list
- presentational complexity of management on user interface side
- very complex SDMX query message would be needed to query for data according to flags

Thus this approach is not recommended to be used. For completeness the table shows a possible implementation of this approach:

Code	Description
A	Normal
B	Time series break
BD	Time series break, Definition differs
BDE	Time series break, Estimated value, Definition differs
etc.	
D	Definition differs
DE	Definition differs, Estimated value
DEP	Definition differs, Estimated value, Provisional value
etc.	
E	Estimated value
EP	Estimated value, Provisional value
etc.	

If further combinations are needed, these can be created on an *ad hoc* basis by selecting the necessary codes from the basic code list and sorting them alphabetically. Inversely, implementers might wish to reduce the list of possible options, would all options above not be necessary for their specific needs.

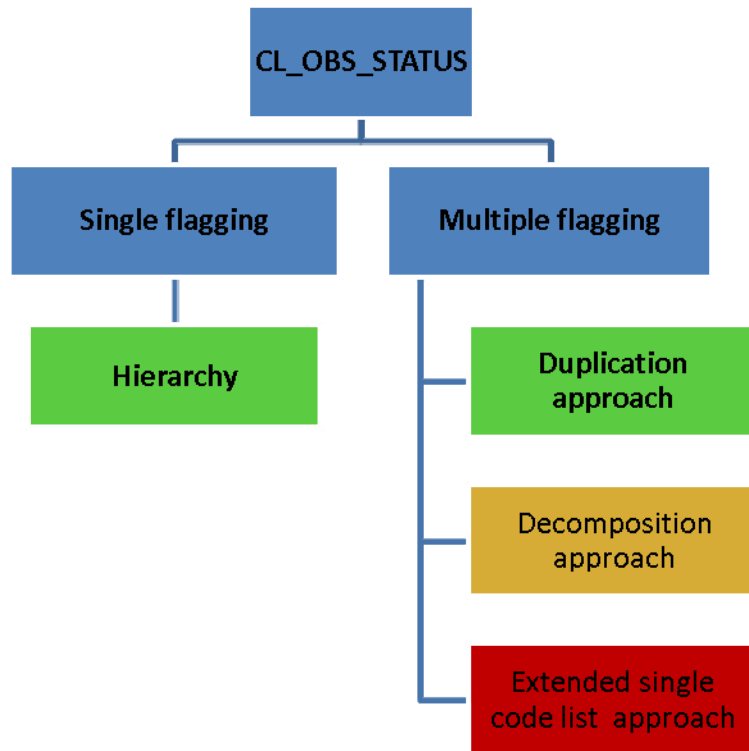
This approach is to be strongly discouraged.

4) Conclusion

From the analysis of the various approaches presented above, it appears clearly that the extended single code list approach cannot be recommended for use.

Although the two remaining approaches, i.e. the duplication approach and the decomposition approach, may qualify for being recommended, it is preferable to give precedence to one approach in order to improve harmonisation across implementations. Considering its relative simplicity in terms of maintenance, the recommended option is option based on the "duplication approach".

5) Synthetic overview of solutions proposed and suggested recommendations



Colour key

- Recommended solution
- Accepted but not preferred solution
- Strongly discouraged