

Developer check list for data elements and data set specifications

Exported from METEOR (AIHW's Metadata Online Registry)

© Australian Institute of Health and Welfare 2024

This product, excluding the AIHW logo, Commonwealth Coat of Arms and any material owned by a third party or protected by a trademark, has been released under a Creative Commons BY 4.0 (CC BY 4.0) licence. Excluded material owned by third parties may include, for example, design and layout, images obtained under licence from third parties and signatures. We have made all reasonable efforts to identify and label material owned by third parties.

You may distribute, remix and build on this website's material but must attribute the AIHW as the copyright holder, in line with our attribution policy. The full terms and conditions of this licence are available at <https://creativecommons.org/licenses/by/4.0/>.

Enquiries relating to copyright should be addressed to info@aihw.gov.au.

Enquiries or comments on the METEOR metadata or download should be directed to the METEOR team at meteor@aihw.gov.au.

Developer check list for data elements and data set specifications

The Developer checklist provides the list of the things that are needed for a comprehensive check of a metadata item against the business rules. Changing from one registration status to another needs checking a specific set of business rules.

Things to be checked for all metadata items;

1. Developer has used Australian English for all the content of the metadata items unless specified otherwise.
2. The first occurrence of the abbreviation is spelled out the word or phrase in full enclosing the abbreviation within parentheses immediately to the right of the word in each metadata item.
 - Exception: The abbreviated classification scheme name included in value domain and data element name.
3. First letter of the first word is capitalised within the name of a publication or organisation.
4. Nouns are expressed in singular form unless the term itself is plural in nature.
5. Identify the item that is being superseded by this new item if applicable.
6. Add a Superseded by relationship between the two.

Things to be checked for an Object class

1. Ensure object class name adheres to [Object class name rules](#).
2. Ensure the definition attribute follows [Object class definition rules](#).

Things to be checked for a Property

1. Ensure property name adheres to [Property name rules](#).
2. Ensure the definition attribute follows [Property definition rules](#).

Things to be checked for a Data element concept

1. Ensure data element concept name does not contradict to [Data element concept name rules](#).
2. Make sure [Object class rules](#) for adding object class to data element concept [Property rules](#) for adding property to data element concept are followed properly.
3. Ensure the definition attribute follows [Data element concept definition rules](#).

Things to be checked for a Value domain

1. Check value domain name complies [Value domain name rules](#).
2. Ensure the definition attribute [Value domain definition rules](#).
3. Make sure [Classification scheme rules](#) for adding classification scheme to value domain are followed properly.
4. Check that representational attributes like representation class, data type, format, unit of measure are filled where they are appropriate for each other.
5. Make sure [Permissible value rules](#) for adding permissible values to value domain are followed properly.
6. Notify Administrator if new unit of measure needs to be added to the list of unit of measures.

Things to be checked for a Data element

1. Check data element name complies with [Data element name rules](#).
2. Ensure the definition attribute follows [Data element definition rules](#).
3. Make sure [Value domain rules](#) for adding value domain to Data element are followed properly.
4. Make sure [Data element concept rules](#) for adding data element concept to data element are followed properly.
5. Include the data element in the respective DSS and include the verification rules and justification, if applied, using [Justification rules](#).