

## 4.6 Other Use Cases

This section includes the following use cases related to forms:

- [4.6.1 Users create forms to collect data for models and data elements that are already registered in the metadata repository.](#)
- [4.6.2 Create reusable modules](#)
- [4.6.3 Import existing form from caDSR form repository into new form tool](#)
- [4.6.4 Generate fillable form](#)

### 4.6.1 Users create forms to collect data for models and data elements that are already registered in the metadata repository.

A Data Entry Form builder is notified of a new study or need for a new clinical data capture form by an end user. The Data Manager/Clinical actor begins with the requirements from the end use, usually an existing form and opens a form editor. The actor can create a new Form. The actor can search for an existing model in the metadata registry and drag it into the palette. The palette self organizes based on the classes and attributes in the model and the existing relations of items in the model. The actor can delete items in the modules, drag attributes into new modules on their form to organize the form according to their anticipated data entry workflow.

If there is not an existing model that the actor can use, the actor browses existing Forms in the local or remote forms registries that could be reused based on search parameters of the form such as the type of study, the type of form (including but not limited to interview, survey, and clinical trial enrollment) or other criteria. If the Data Manager/Clinical Actor is unable to locate an existing form, the actor may search for Form Modules in the local or remote Forms registry to construct a new form.

If the actor finds a form or a module the actor can reuse, the actor drags the existing form or module to the build palette to start the form project. This is repeated for each section identified by the requirements. When a template is present, the modules self organize according to the template. If a template does not exist, the user connects the modules together with the guidance of the system on the existing classes of the data elements within the modules and relations between them.

If there is not an existing module that the actor can use, the actor can search for possible questions in the metadata repository that meet the questions on their existing form. Search is enhanced by semantic infrastructure to quickly allow users to search for questions by entering text of the question, keywords or values in their value set. The system presents the user with choices the user can put into the forms palette or temporary area once the user has found all the questions needed to create the form.

The selection of a form, module or question on the palette exposes the attributes of the item in a window where the actor may continue curation.

Selection of the permitted value set is done by matching the semantics of the question in the module to the metadata registry to identify the permitted value set for that attribute. If no value set exists the user is passed to the Common Terminology Services (CTS) service to construct a value set.

If the actor is reusing a class/attribute for which a value set has been specified (in current system at the Data Element Concept (DEC) level), the system shows the actor the value set, existing value domains that have already be paired with the DEC, and the actor can choose to reuse an appropriate value domain, or choose to create a new one by selecting values from the value set.

The actor can optimally choose to search for a new value set by keying in the values from the answer list for the forms question and the system will see if a value set already exists.

The actor adds metadata to the form package and the system checks to see if the questions in the module have been created before with the same value sets, that the system reuses or creates new data element identifiers for each of the form components requiring new identifiers.

The actor can rearrange the permissible values to the preferred order for the form. The actor next defines the rules for skip patterns and form flow logic that is stored with the form. The form is then identified and its associated metadata is passed to the metadata registry. The metadata registry notifies subscribers of the availability of a new form type.

The actor can download an XML description of the form or save the form to a cart for later review or pulling into a data entry system. Software Developer actors can access the form metadata through services to retrieve the form and its associated metadata for implementation in a local system.

### 4.6.2 Create reusable modules

A Data Entry Form builder is notified of a new study or need for a new clinical data capture form by an end user. The Data Manager/Clinical actor begins with the requirements from the end user, usually an existing form, and opens a form editor. The Data Manager/Clinical actor selects to create a new Module. The actor can search for an existing model, form, module or question in the metadata registry and drag it into the palette.

The actor can choose to have the palette self organize, or to apply manual organization based on the existing form and workflow. If the actor chooses self organize, the questions in the module are organized based on information in the metadata registry related to the questions in the model. If the attribute is found in several models, the actor can chose the model to apply the relations after inspecting the matching models.

The actor can delete items in the modules, and drag attributes into new modules on the form to organize the form according to anticipated data entry workflow.

Selection of the permitted value set is done by matching the semantics of the question in the module to the metadata registry to identify the permitted value set for that attribute. If no value set exists the user is passed to the CTS service to construct a value set.

If the actor is reusing a class/attribute for which a value set has been specified (in current system at the DEC level), the system shows the actor the value set and existing value domains that have already be paired with the DEC. The actor can choose to reuse an appropriate value domain, or choose to create a new one by selecting values from the value set.

The actor can optimally choose to search for a new value set by keying in the values from the answer list for the forms question and the system will see if one already exists.


The actor adds metadata to the form package and the system checks to see if the questions in the module have been created before with the same value sets, that it reuses or creates new data element identifiers for each of the form components requiring new identifiers.

The actor next defines the rules for skip patterns and form flow logic that is stored with the form. The form is then identified and its associated metadata is passed to the metadata registry.

The metadata registry notifies subscribers of the availability of a new form type. The actor can download an XML description of the form or save the form to a cart for later review or pulling into a data entry system.

Software Developer actors can access the form metadata through services to retrieve the form and its associated metadata for implementation in a local system.

#### 4.6.3 Import existing form from caDSR form repository into new form tool

The Data Manager/Clinical Form Builder actor searches the existing repository for a form by name, identifier or keywords. The actor pulls the existing form into the form palette. The palette self organizes based on the existing form structure to match to the new form structure (for example, Form to a Form Project, Module to a Common Message Element Type (CMET) ).

The actor can add or delete content from the copied based on need. The actor can reorganize the content based on need.

The actor adds metadata to the form package and the system checks to see if the questions in the module have been created before with the same value sets, that the system reuses or creates new data element identifiers for each of the form components requiring new identifiers.

The actor next defines the rules for skip patterns and form flow logic that is stored with the form. The form is then identified and its associated metadata is passed to the metadata registry.

The metadata registry notifies subscribers of the availability of a new form type. The actor can download an XML description of the form or save the form to a cart for later review or pulling into a data entry system.

Software Developer actors can access the form metadata through services to retrieve the form and its associated metadata for implementation in a local system.

#### 4.6.4 Generate fillable form



##### Note

This may also be captured elsewhere in the Semantic Infrastructure 2.0 Roadmap.

This use case extends all others. The actor can use the metadata exported from the registry to automatically generate a fillable form that follows all the rules expressed in the form project.