Workflow Utilities

Page Contents

Workflow Utilities

If an asset collection you are working with has any workflow in progress you can access them by clicking on **Workflows** tab for the collection.

If you are responsible for transitioning a workflow to the next state, you will also see it listed in the **My Workflows** page that you can access directly from the left Navigation Bar.

In both cases, you will have up to two options:

- Go to Workflow - this will take you to a page where you can see the current state of the workflow and will show a diagram depicting its state transitions. It is called Workflow Status and Transition page.
- Go to Asset - this options is only available for workflows that pertain to one specific asset. Clicking on this button will take you directly to the assets's form

On top of the page you will see the same tabbed horizontal navigation menu as you see when working directly with an asset collection, but its color will be pale yellow instead of grey. This will give you a visual indication that you are in a workflow.

Any changes you will make will stay in the working copy associated with this workflow and will not be visible outside it.

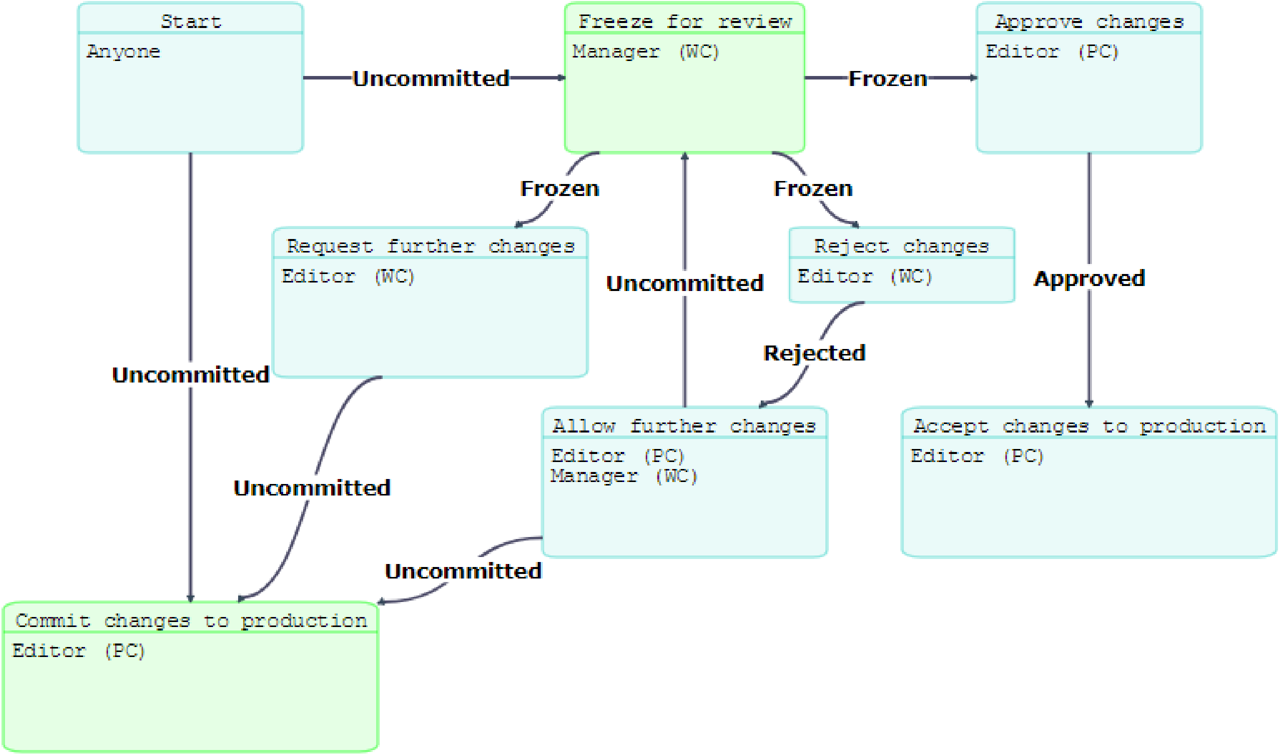
Workflow Status and Transition Tab

This is the first tab in the horizontal navigation menu. It has a gear icon and contains a name of the workflow and its description. If you have manager privileges you will be able to edit either one by clicking on the text, editing, and confirming by clicking on the edit-box's check-mark.

The page shows shows the current state of the workflow and a diagram depicting its state transitions. This page also lets a user transition the workflow to the next state. Depending on the current state and the role you play in the workflow process, you may see action buttons for moving the workflow to the next state.

The Basic Workflow

The standard EDG workflow template includes the *Basic workflow*, which supports optional review cycles and staging before committing changes to production.



The Basic workflow's states and actions (transitions)

Basic WF Status	Action	Role(s): Graph(s) *	Description	Resulting Status
Cancel this workflow	Manager: either	Irrevocably deletes the entire WF	*N/A*	
Uncommitted (editable)	Commit changes to production	Editor: Production	Applies all WF changes into production	Complete
	Freeze for review	Manager: WF	Pauses WF editability	Frozen for Review
Frozen for Review	Request further changes	Editor: WF	Resumes WF editability	Uncommitted
	Reject changes	Editor: WF	Blocks applying any WF changes into production	Rejected
	Approve changes	Editor: Production	Ends WF editability (staged for production)	Approved
Rejected	Allow further changes	Editor: Prod. & Manager : WF	Resumes WF editability	Uncommitted
Approved	Accept changes to production	Editor: Production	Applies all WF changes into production	Complete

Complete				Complete
* All actions require at least viewer permissions on both the production and WF copies.				

View and Edit Page

The next tab in the horizontal menu opens the Workflow's editing view. The name of the tab varies depending on the type of collection.

Editing works exactly the same way as if you were editing production copy of the Workflow except that all changes are kept in a working copy until the workflow completes successfully. (See Workflows for details on the difference between using workflows and directly editing production copies.)

Settings View

The Settings view provides access to asset collection's metadata. It also lets you view and modify asset collections that are included in this collection.

NOTE: Workflow Settings are read-only. They can be modified on the production version by editors.

Included-By and Includes

These sections show the references (owl:imports) from and to other collections. Users with editor privileges for this Workflow can modify its includes. The choice of collections-to-include is restricted to collection types that are either *required* or *permitted* for the current Workflow. See the main Workflows page (under Home > Create) for additional information.

Default Namespace

The default namespace is used to construct URIs (unique identifiers) for the classes, properties and other resources in the Workflow. Changes to it will not impact URIs of already existing resources.

Graph URI

This is an internal identifier for any asset collection created by EDG. It can't be changed.

External Graph URI

This option defines a URI that automatically maps with EDG's internal Graph URI during imports and exports. RDF file import operation will automatically redirects any owl:imports statements in a file to the asset collections in EDG with external graph URIs that match those in the file's owl:imports statements. The inverse mapping happens when graphs are exported back to RDF files: their external Graph URI is used instead of the internal urn:x-evn-master:... URIs.

Metadata

This section lets you view or edit information about the Workflow. There is a rich selection of metadata fields, and it is easy to configure EDG to include additional fields if required. The metadata is organized into sections. The view mode only shows the sections and fields that contain information, while the edit mode shows all sections and all fields.

Overview

This section records descriptive properties of the Workflow, including its official name (if different from the common name or label).

subject area

This is the governance area (business or data-subject) to which this Workflow is assigned (if any).

Status

The **Status** section records the life cycle stages of the Workflow.

Users View

Workflow Permissions

For any asset collection and a working copy of a collection in a context of a workflow, access to its basic viewing, editing, and utility functions is controlled by the three nested permission profiles: *viewer*, *editor*, and *manager* (where each profile is a superset of the preceding one). On each collection, these permissions are set for various users, either as individuals or as security roles (e.g., from LDAP).

Workflow Permissions section list users that have access to this specific Workflow or, in the context of a workflow, its working copy. A *blank* setting indicates no access for the user or role.

Users with manager privileges can use this page to assign permission profiles to other users. For non-managers, this view is read-only.

In a context of a production copy, permissions can be set for any user. In a context of a workflow permissions can be set only for users that already have (at least) a viewers privileges for the production copy. A given user can have different permission profiles for the collection itself (production copy) and one of its workflow working copies. Similarly, they could have different permissions in the context of different workflows.

For example, if Jane has *manager* permission profile for an asset collection, she can give John *viewer* privileges for it and *editor* privileges for one of its workflow copies. John can then make changes in the context of a workflow.

A user that has any governance role for a collection is automatically granted viewer privileges for the collection and any of its workflows.

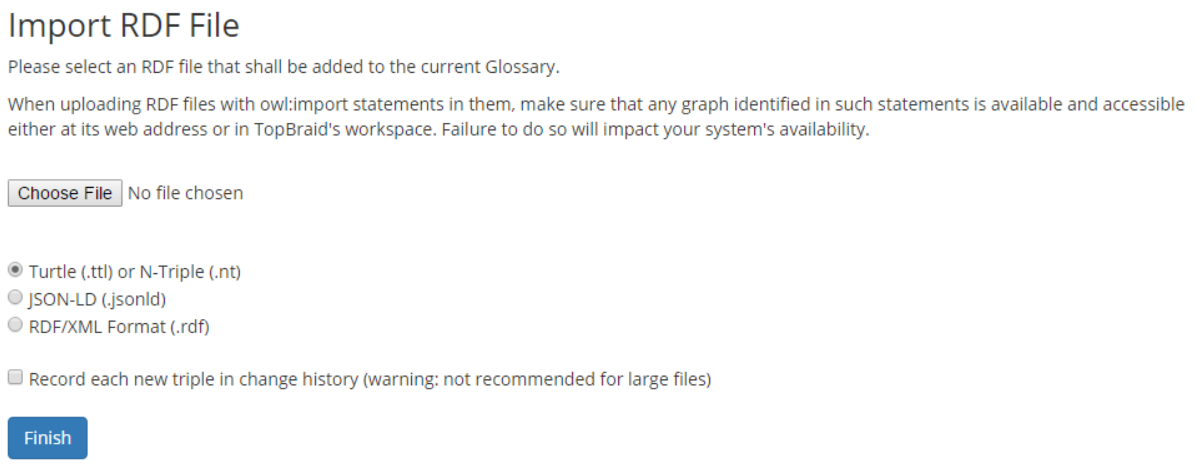
A user with multiple permission profiles on a given collection or workflow receives the greatest level assigned. See Workflows - Rights Entailed by Permissions Profiles for details.

Import View

From any Workflow's production or working-copy home page, the **Import** functions lets you load data into the given Workflow from external sources such as RDF files, spreadsheets, etc. You will only see the Import View and its operations if you have edit privilege for the asset collection.

Import RDF File

Any Workflow can import data from an external RDF file (in a serialized format). The **Import > Import RDF File** link shows a screen where the **Choose File** button opens a dialog for picking the external source file.



Import RDF File

Please select an RDF file that shall be added to the current Glossary.

When uploading RDF files with owl:import statements in them, make sure that any graph identified in such statements is available and accessible either at its web address or in TopBraid's workspace. Failure to do so will impact your system's availability.

Choose File No file chosen

● Turtle (.ttl) or N-Triple (.nt)
○ JSON-LD (.jsonld)
○ RDF/XML Format (.rdf)

□ Record each new triple in change history (warning: not recommended for large files)

Finish

Choose the source file and **identify its format**. Decide whether to record new triples in the change history (use with care!) and then click **Finish** to complete the import. A message will indicate whether the import was successful.

Note that if your file contains any schema definition (classes, properties, shapes), you will only be able to import it into an Ontology. If you have both instances and schema data in one file, either split the file before import or follow instructions in Refactoring Ontologies.

When importing RDF files into an Ontology or into a Taxonomy, EDG will perform some transformations:

- For ontologies, "subclass of Thing" statements will be added for classes that have no parents. This is done to ensure that these classes are visible in the Class Hierarchy.
- For taxonomies, "narrower concept" relationships will be used to generate inverse "broader concept" relationships. This is done to ensure that such concepts are visible in the Concept Hierarchy.

When importing RDF into a Working Copy, the addition of each triple can be added as an entry in the change history, where it will be available to all the relevant reports. When importing into a production copy, the Record each new triple in change history checkbox gives you the option of adding these to the change history; note that this is **not** recommended when importing large amounts of data.

Import Spreadsheet using Template

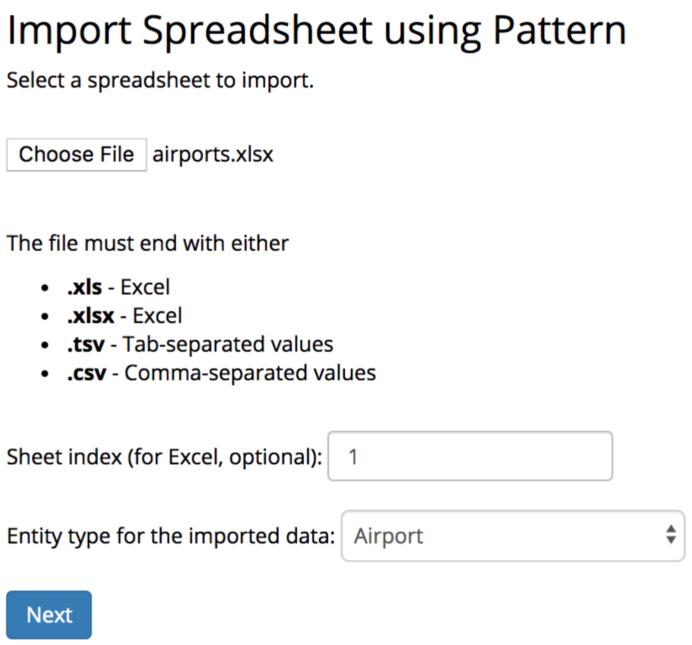
This screen lets you pick a spreadsheet and a template that will be used to convert the spreadsheet data into reference data. The template may be created using the mapping process explained in the *...using Pattern* section, below.

The mapping can also be created using TopBraid Composer when the simple mapping described above is insufficient and you need to perform more complex transformations–for example, concatenation of values. TopBraid Composer's SPINMAP tool provides a drag-and-drop interface that makes it especially easy to create more complex mappings.

Templates developed with TopBraid Composer must be stored in files with ".tablemap." in their name (for example, myMapping.tablemap.ttl) and be uploaded to the EDG server to be available to EDG users. Spreadsheet imported using a template must have exactly the same structure as the spreadsheet used to develop the template. The names and order of the columns must be exactly the same. If multiple worksheets are used, the order and structure of each worksheet (even for worksheets that are not imported) must be the same.

Import Spreadsheet using Pattern

This link shows the following screen:



Import Spreadsheet using Pattern

Select a spreadsheet to import.

airports.xlsx

The file must end with either

- **.xls** - Excel
- **.xlsx** - Excel
- **.tsv** - Tab-separated values
- **.csv** - Comma-separated values

Sheet index (for Excel, optional): 1

Entity type for the imported data: Airport

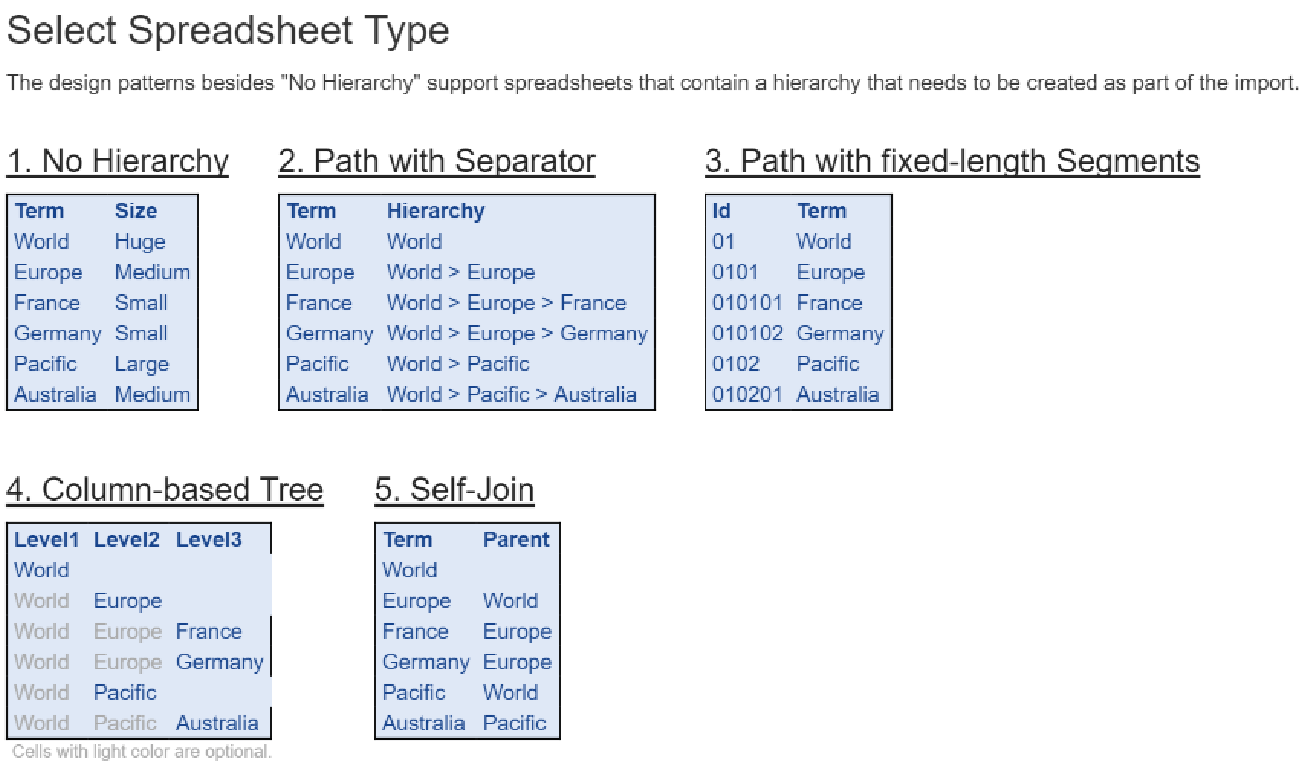
Click Choose File to pick the external spreadsheet file whose data you want to import into the Workflow. This may be an Excel file (file-type extensions: *.xls* or *.xlsx*), a tab-separated value (*.tsv*) file, or a comma-separated value (*.csv*) file. The file-name should have the expected extension. Because an Excel file may have more than one sheet of data, this screen lets you specify a **sheet index** value to identify which sheet to read in. The default is 1, for the first sheet.

The sheet index counts all sheets in an Excel workbook, including hidden ones. For example, if you enter a 3 here and EDG seems to import the second sheet, there may be a hidden one between the first and second sheet that made the third one look like it was the second one when Excel was displaying the workbook. The Excel online help explains how to check for the existence of hidden sheets.

The Entity type for the imported data field identifies what assets you will be importing. Each row in the spreadsheet will be brought into EDG as an instance of the selected *class* and you will be able to map spreadsheet columns to the declared properties of the class. Make the (1) file, (2) sheet index, and (3) entity type (class) selections and click Next .

Select Spreadsheet Type

This view enumerates five possible (column-wise) layout patterns for the spreadsheet's row-item data, showing an example of each pattern. For data explicitly structured into a hierarchy, like a taxonomy, there are four layout options. For all other data, there is the *No Hierarchy* layout (#1). In the hierarchical layouts, each row item also indicates its hierarchical path, either explicitly (absolute path, #2, #3, #4) or implicitly (recursive path, #4, #5; note that lighter text in the layout patterns indicates optional data).



Select Spreadsheet Type

The design patterns besides "No Hierarchy" support spreadsheets that contain a hierarchy that needs to be created as part of the import.

1. No Hierarchy

Term	Size
World	Huge
Europe	Medium
France	Small
Germany	Small
Pacific	Large
Australia	Medium

2. Path with Separator

Term	Hierarchy
World	World
Europe	World > Europe
France	World > Europe > France
Germany	World > Europe > Germany
Pacific	World > Pacific
Australia	World > Pacific > Australia

3. Path with fixed-length Segments

Id	Term
01	World
0101	Europe
010101	France
010102	Germany
0102	Pacific
010201	Australia

4. Column-based Tree

Level1	Level2	Level3
World		
World	Europe	
World	Europe	France
World	Europe	Germany
World	Pacific	
World	Pacific	Australia

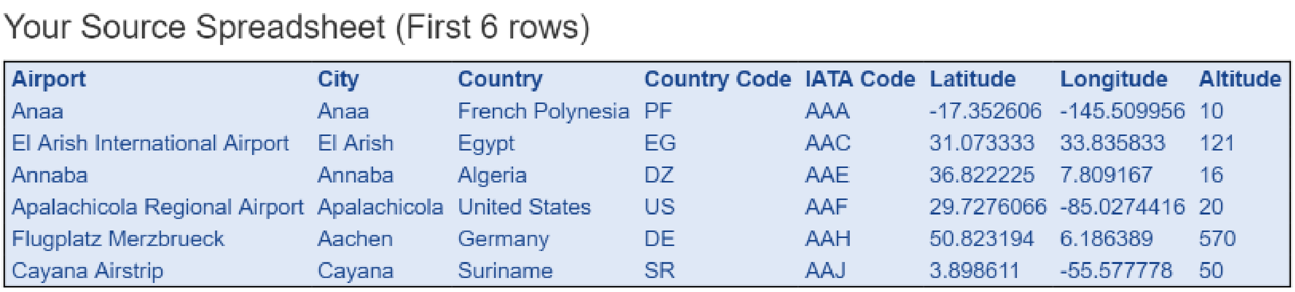
Cells with light color are optional.

5. Self-Join

Term	Parent
World	
Europe	World
France	Europe
Germany	Europe
Pacific	World
Australia	Pacific

Note the *header row* of column labels in every layout. The imported spreadsheet should have such a header row.

Below the five layout options, the view shows a sample of the spreadsheet's actual data. This following image shows a spreadsheet of airport codes.



Your Source Spreadsheet (First 6 rows)

Airport	City	Country	Country Code	IATA Code	Latitude	Longitude	Altitude
Anaa	Anaa	French Polynesia	PF	AAA	-17.352606	-145.509956	10
El Arish International Airport	El Arish	Egypt	EG	AAC	31.073333	33.835833	121
Annaba	Annaba	Algeria	DZ	AAE	36.822225	7.809167	16
Apalachicola Regional Airport	Apalachicola	United States	US	AAF	29.7276066	-85.0274416	20
Flugplatz Merzbrueck	Aachen	Germany	DE	AAH	50.823194	6.186389	570
Cayana Airstrip	Cayana	Suriname	SR	AAJ	3.898611	-55.577778	50

Select the **layout title** link that most correctly corresponds to the chosen spreadsheet's structure.

Import Spreadsheet

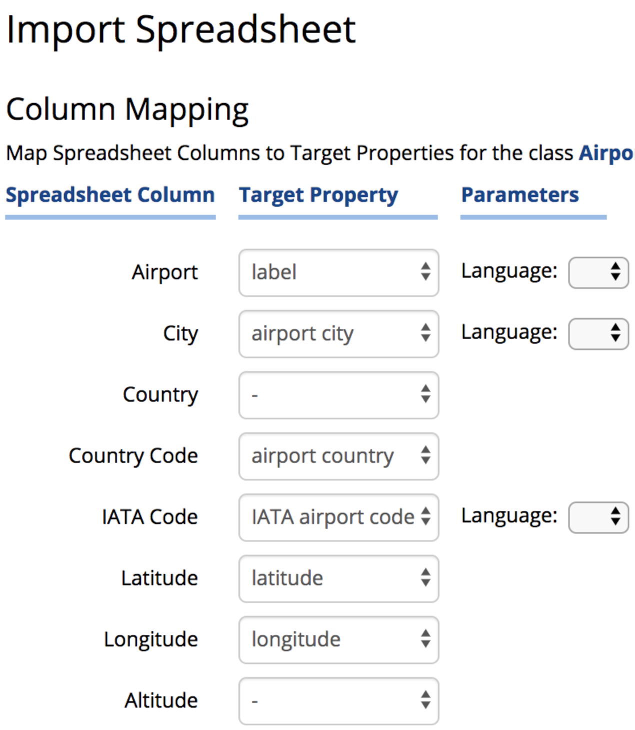
This view defines the data-mapping rules from the spreadsheet's columns into the class and optional hierarchy structures chosen for the collection data. It also lets users to (1) preview the import, (2) save its settings as a template for future imports, and it (3) initiates the import processing.

Column Mapping

The *Column Mapping* settings specify which spreadsheet columns correspond to which properties in the target Workflow. Typically, the target properties belong to the target entity type (class or asset type) you have selected on the previous page. When column names are close to the property names, EDG will automatically propose the mapping.

The mapping also supports *inverse* relationships *to* the target entities. Any unmapped columns will be ignored during the import, their data will not be imported.

The following example shows a mapping page presented for the "No Hierarchy" pattern.

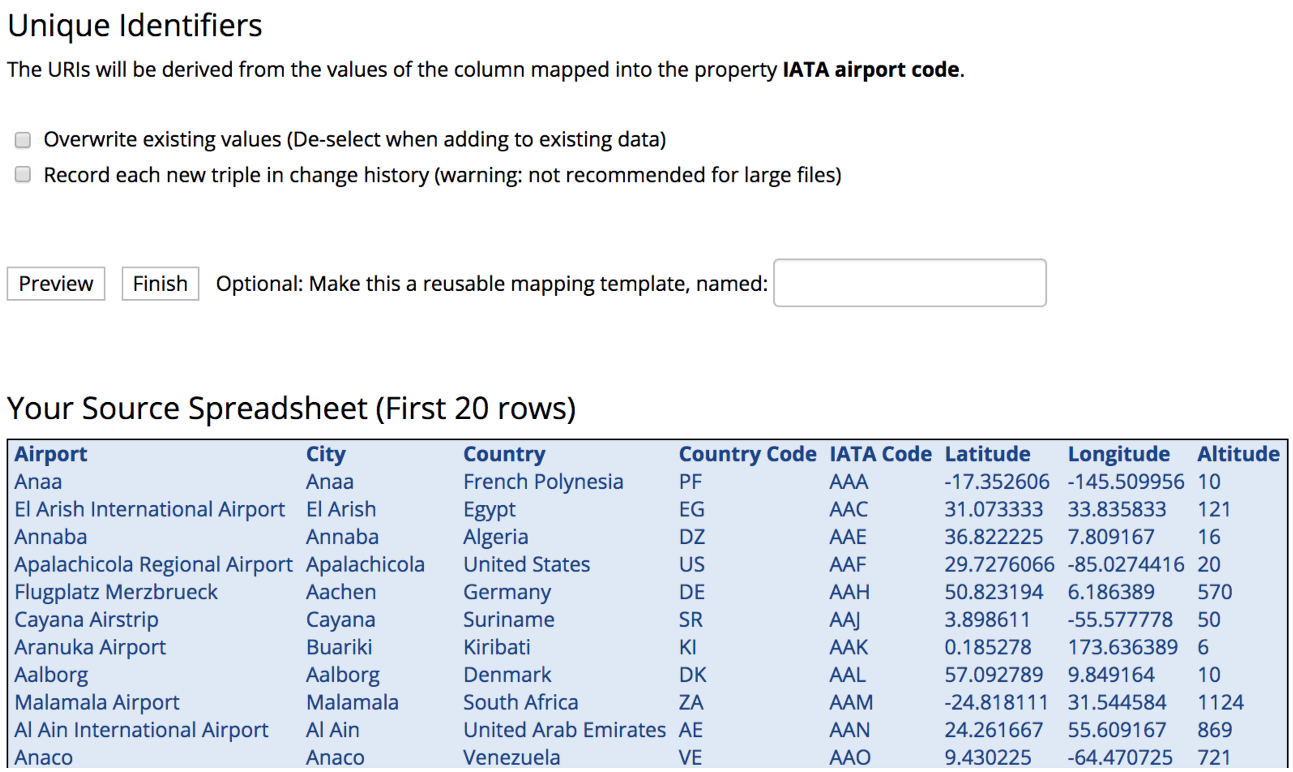


Import Spreadsheet

Column Mapping

Map Spreadsheet Columns to Target Properties for the class **Airport**:

Spreadsheet Column	Target Property	Parameters
Airport	label	Language:
City	airport city	Language:
Country	-	
Country Code	airport country	
IATA Code	IATA airport code	Language:
Latitude	latitude	
Longitude	longitude	
Altitude	-	



Unique Identifiers

The URIs will be derived from the values of the column mapped into the property **IATA airport code**.

□ Overwrite existing values (De-select when adding to existing data)

□ Record each new triple in change history (warning: not recommended for large files)

Optional: Make this a reusable mapping template, named:

Your Source Spreadsheet (First 20 rows)

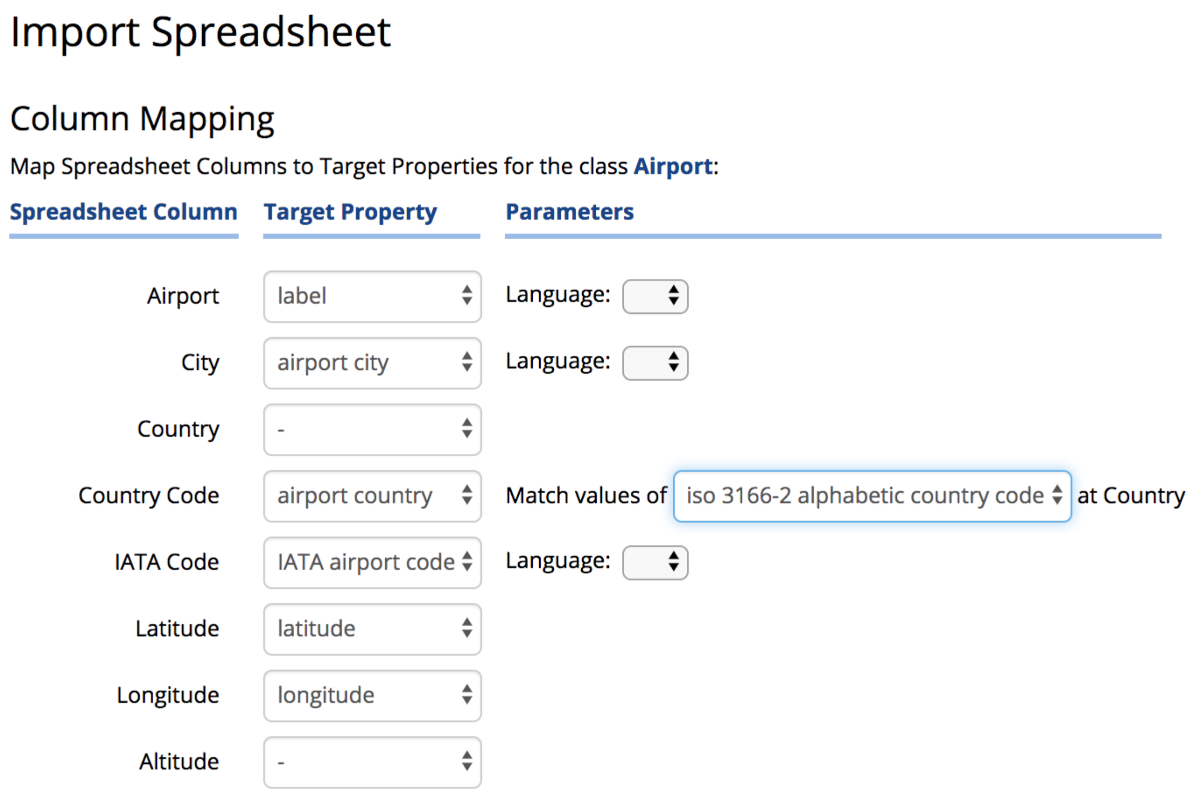
Airport	City	Country	Country Code	IATA Code	Latitude	Longitude	Altitude
Anaa	Anaa	French Polynesia	PF	AAA	-17.352606	-145.509956	10
El Arish International Airport	El Arish	Egypt	EG	AAC	31.073333	33.835833	121
Annaba	Annaba	Algeria	DZ	AAE	36.822225	7.809167	16
Apalachicola Regional Airport	Apalachicola	United States	US	AAF	29.7276066	-85.0274416	20
Flugplatz Merzbrueck	Aachen	Germany	DE	AAH	50.823194	6.186389	570
Cayana Airstrip	Cayana	Suriname	SR	AAJ	3.898611	-55.577778	50
Aranuka Airport	Buariki	Kiribati	KI	AAK	0.185278	173.636389	6
Aalborg	Aalborg	Denmark	DK	AAL	57.092789	9.849164	10
Malamala Airport	Malamala	South Africa	ZA	AAM	-24.818111	31.544584	1124
Al Ain International Airport	Al Ain	United Arab Emirates	AE	AAN	24.261667	55.609167	869
Anaco	Anaco	Venezuela	VE	AAO	9.430225	-64.470725	721

When importing *attribute* values, the spreadsheet column should contain cell-values that matching the datatype of the target attribute. If, as part of the mapping, you select an optional **Language** value, import will add the selected language tag.

When importing *relationships* EDG needs to understand which existing resources to build a link to. Alternatively, it can create a URI as a value of the relationship even if a resource with the URI does not yet exist in the asset collection. After you select a relationship from the dropdown, you will have the following methods for directing EDG on how to build relationship values:

1. If the related class has a designated *primary key (PK)* constraint, no additional information is needed. EDG will use the data in the mapped column to form the URIs of related resources according to the primary key definition. This means that if the related class has a primary key defined, values in the column must correspond to the property used as the key. For example, in the screenshot above Airport has an "airport country" relationship to Country and Country has a primary key that uses 2 character country codes. Thus, we mapped the Country Code column in the spreadsheet to the "airport country" relationship and did not need to provide any additional information.
2. If all the values in the column are recognized as valid URLs, then they can be used as-is as URIs of related resources, as indicated by the associated **Use values as URIs** label.
3. Otherwise, you will see an option to select a property of the related class to match on. This property will be used to find resources at the end of the relationship. For example, if Country did not have a primary key, we could map either Country or Country Code to the "airport country" relationship. If we mapped the Country column, we would then select the "short name" as the property which values to use to identify the related resources. If we mapped the Country Code column, we would select "ISO 3166-2 alphabetic country code" as the property which value to use to identify the related resources. Unlike the first two options, this method will only create a relationship if it finds already existing resource it can connect to.

Option 3 is demonstrated in screenshot below by removing the primary key from the class Country.



Import Spreadsheet

Column Mapping

Map Spreadsheet Columns to Target Properties for the class **Airport**:

Spreadsheet Column	Target Property	Parameters
Airport	label	Language:
City	airport city	Language:
Country	-	
Country Code	airport country	Match values of iso 3166-2 alphabetic country code at Country
IATA Code	IATA airport code	Language:
Latitude	latitude	
Longitude	longitude	
Altitude	-	

For *inverse* relationships, the spreadsheet column represents some other class's *reference to* the imported entities. Similar to forward (non-PK) relationships, if an inverse relationship is the chosen mapping, then there is a further choice of which referring-class property to use to identify the referring instances.

Even if the referenced property is *not* designated as a primary key (PK), it is still assumed that all of the corresponding property-values are unique across all referenceable instances. If duplicate values exist, then the referenced instance will be assigned arbitrarily.

Also note that if the related class designates a primary key property or if column values are URLs, then imported rows will *always* construct a reference, regardless of whether referenced resources exist. On the other hand, if there is **no** such PK designation, then imported rows will construct a reference *only if* a matching instance exists.

As explained below, if the target of a relationship has the same entity type as the entity type for the imported data AND you are using matching on the property values to build a relationship, the ***Override existing values*** option must be unchecked. Otherwise, the relationship will not be created.

If the imported rows will create any new instances, as opposed to only adding data to existing ones, then one of spreadsheet columns should be mapped to the *label* property, which is used as the display name.

When importing into *reference datasets*, then one of spreadsheet column must map to the primary-key property that is designated for the dataset's *main entity* class. For example, the screen image above identifies this field as the IATA code.

If the imported rows are adding new data values to existing instances and/or adding new instances, it is best to **uncheck** the *Override existing values* option. Checking this option has the following consequences:

- If an instance already exists and has a value for any of the mapped columns, the value will be replaced with data coming from a spreadsheet.
- Relationships between instances of the same type that rely on matching of values will not be created (because these values may be overridden as part of the processing).
- When working with Taxonomies, a combination of checked *Override existing values* and the *No Hierarchy* pattern will always make imported instances top concepts of a new Concept Scheme, even if they already exist in the Taxonomy and have parent concepts.

Hierarchical spreadsheet types

If a hierarchical pattern was selected, then there will also be *Hierarchy* settings that specify how the spreadsheet represents the hierarchical relationships of its data items.

Note that one could still use the **No Hierarchy** import pattern for a taxonomy. This would import spreadsheet terms as separate concepts without connecting them hierarchically unless you can indicate some property to use to find broader concepts. The taxonomy's editors could then use the Concept Hierarchy view to manually arrange the imported nodes in the taxonomy's concept tree.

- For **Path with Separator** spreadsheets, make sure to assign at least one column under **Column Mapping** as the preferred label. For other types, the importer can often infer the preferred label column. (See below about the Preview button.)
- For **Column-based Trees** and **Column-Pair Based Tree** spreadsheets, specify the top and bottom levels of the hierarchy by picking the first and last column names.
- For **Path with fixed-length Segments** spreadsheets, specify the column with the path values used as IDs and the length of the segments within the path IDs. In the **Path with fixed-length Segments** sample layout on the Select Spreadsheet Type screen, the Id column has the path values, and each two-digit segment of these values indicates a step of the hierarchy; removing the last two digits of any of those Id values shows the Id value for that term's parent. For example, Australia has an Id value of 010201, which has a parent value of Pacific (Id value 0102), which has a parent of World (Id value 01).
- For **Path with Separator** spreadsheets, in which a spreadsheet entry such as "World > Europe > France" indicates the hierarchical structure above the term "France", specify the column storing these values using the Column containing the paths field and the Path separator character in the field with that name. If your spreadsheet also includes an ID column, the **Hierarchy** section includes a dropdown field to indicate this.
- For **Self-Join** spreadsheets, there are columns to specify the **Column containing the parent ids** and the **Column containing the child ids**. In the Self-Join sample on the Select Spreadsheet Type screen, these would be the Parent and Term columns, respectively. A **Hierarchy Property** field also lets you set whether you want the "has broader" property used to identify relationships in the displayed hierarchy or something else.
- The **Generate in inverse direction** checkbox will reverse the direction of how the property specified in **Hierarchy property** is applied.

Unique Identifiers

This section defines the URI for each imported row. Selecting Overwrite existing values will *delete* an existing value for a mapped property before adding its new (different) value; otherwise, new values will be added to existing ones. Selecting Record each new triple in change history (warning: not recommended for large files) prevents EDG from recording the addition of each new triple in the change history.

A Preview button on the Import Spreadsheet form lets you see the RDF triples that would be generated with the currently configured settings. The browser's *Back* button returns to the form.

The Optional: Make this a reusable mapping template field lets you save all of the settings on this form so that you can later import other spreadsheets with a similar layout without filling out this form. Once you use this field to name a template of settings, if you later pick Import Spreadsheet using Template on the Import tab instead of Import Spreadsheet using Pattern, you'll see a drop-down list of the saved template names, a Browse button to pick a spreadsheet, and a Finish button to perform the actual conversion.

When you are satisfied with the sample data shown on the preview screen, click the Finish button. EDG will display a message about successful importing of the data along with a set of data that you can use as a mapping file for imports of spreadsheets with a similar structure in the future.

Export View

Export View provides access to various export options. You can also export data directly from the pages that let you view and edit Workflow, but this view provides more options. While the primary use of this tab is for running exports, it offers full query access via SPARLQ and GraphQL. This means that some of the utilities in this view will also let you modify data.

Export Workflow as a Graph

This operation lets you export collection's data in a standard RDF serialization format.

Creation of reports and exporting of data are available when working with both production and working copies of reference data to anyone with who has read access.

To export subsets of Workflow data according to custom criteria and sorting, note that EDG's Search screen provides fine-grained control over the data to display on the Search Results area. That form's gear menu offers several choices to export the results into spreadsheet-compatible formats (e.g., for Excel). See Workflow View or Edit for details on **searching**.

For any collection, to generate an RDF serialization of its contents, select Export > Export Workflow as a Graph and then an RDF format: JSON-LD, N-Triples, RDF/XML, Turtle or **TriG**, with or without inferences ("with inferences" adds a dedicated graph named `urn:x-topbraid:inferences`, which has any triples inferred via SHACL or SPIN rules; note that this computes on-the-fly and might be very slow).

Browser interactions might vary: displaying the data directly or via a kind of *view source* command. Alternatively, instead of directly clicking a format link, the browser might provide (on the link) a right-click menu option to *save the link target to a file* (i.e., without explicitly displaying the link result in the browser). A dialog box will prompt for the file location and name.

SPARQL Endpoint

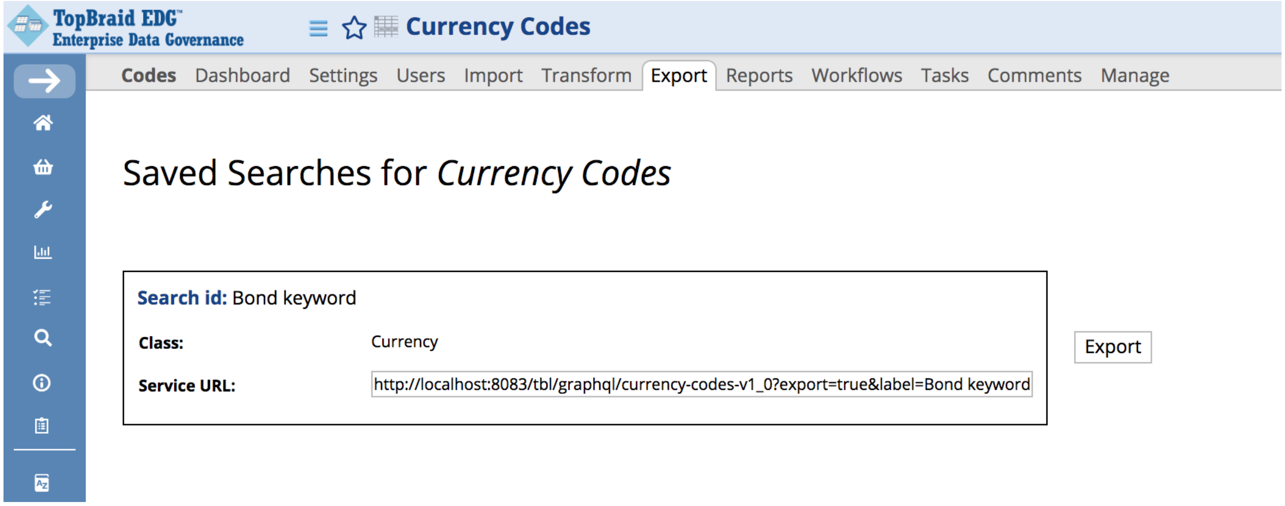
This allows users of the collection to run new or saved SPARQL queries on it and to optionally save queries for others. Saved queries can be deleted by their creators and by collection managers. If SPARQL updates have been enabled by an administrator, editors (and managers) can run them, but viewers cannot. Note that the *Pivot Table* and *Geo* functions can be slow on some platforms and are not supported for Internet Explorer.

Export using Saved SPARQL Query

This lists the SPARQL queries that have been saved for the collection. For each query, it provides a URL that will run it, along with an **Export Query** button that runs it and shows the results.

Export using Saved Search

The Saved Search link shows a screen listing your saved searches. These are searches that you have saved using the **Save search** option in the view/edit page.

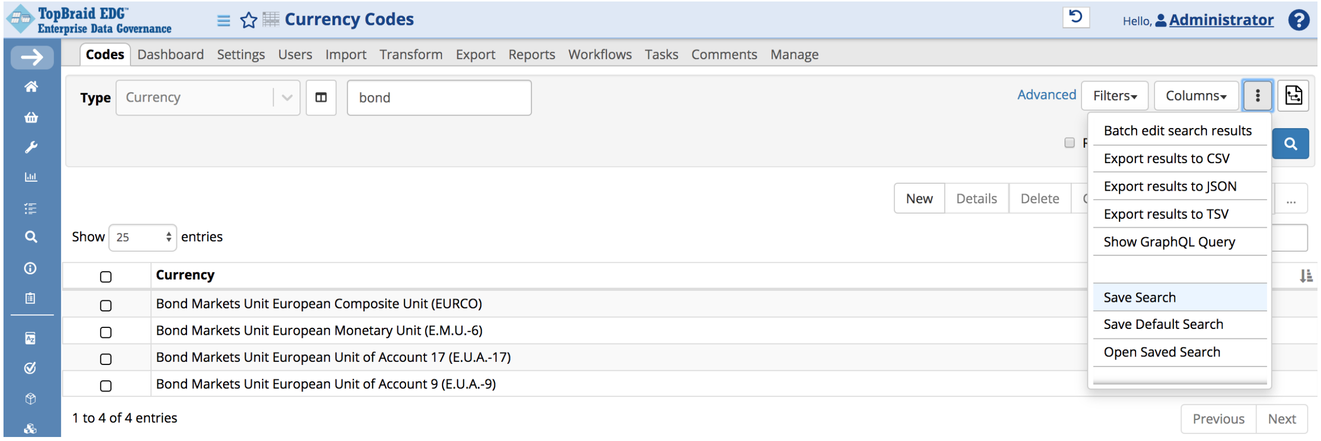


Saved Searches for *Currency Codes*

Search id: Bond keyword

Class: Currency

Service URL: http://localhost:8083/tbl/graphql/currency-codes-v1_0?export=true&label=Bond keyword



Type: Currency

bond

Show 25 entries

	Currency
□	Bond Markets Unit European Composite Unit (EURCO)
□	Bond Markets Unit European Monetary Unit (E.M.U.-6)
□	Bond Markets Unit European Unit of Account 17 (E.U.A.-17)
□	Bond Markets Unit European Unit of Account 9 (E.U.A.-9)

1 to 4 of 4 entries

Clicking Export button will download the search results. Your saved searches are web services. They can also be used as an APIs by other systems.

Reports View

Reports View for Workflow lets you generate various standard reports for it. Custom reports are also possible.

Problems and Suggestions Report

For any Workflow (production copy or working copy managed by a workflow), **Reports > Problems and Suggestions** checks the current state of the Workflow against all of its applicable quality rules (i.e., its shapes and validity constraints they define) and enrichment rules. A message box shows the rule-processing progress and then shows the report. Note that the report results are also reflected in the **Dashboard > Completeness and Validity** display.

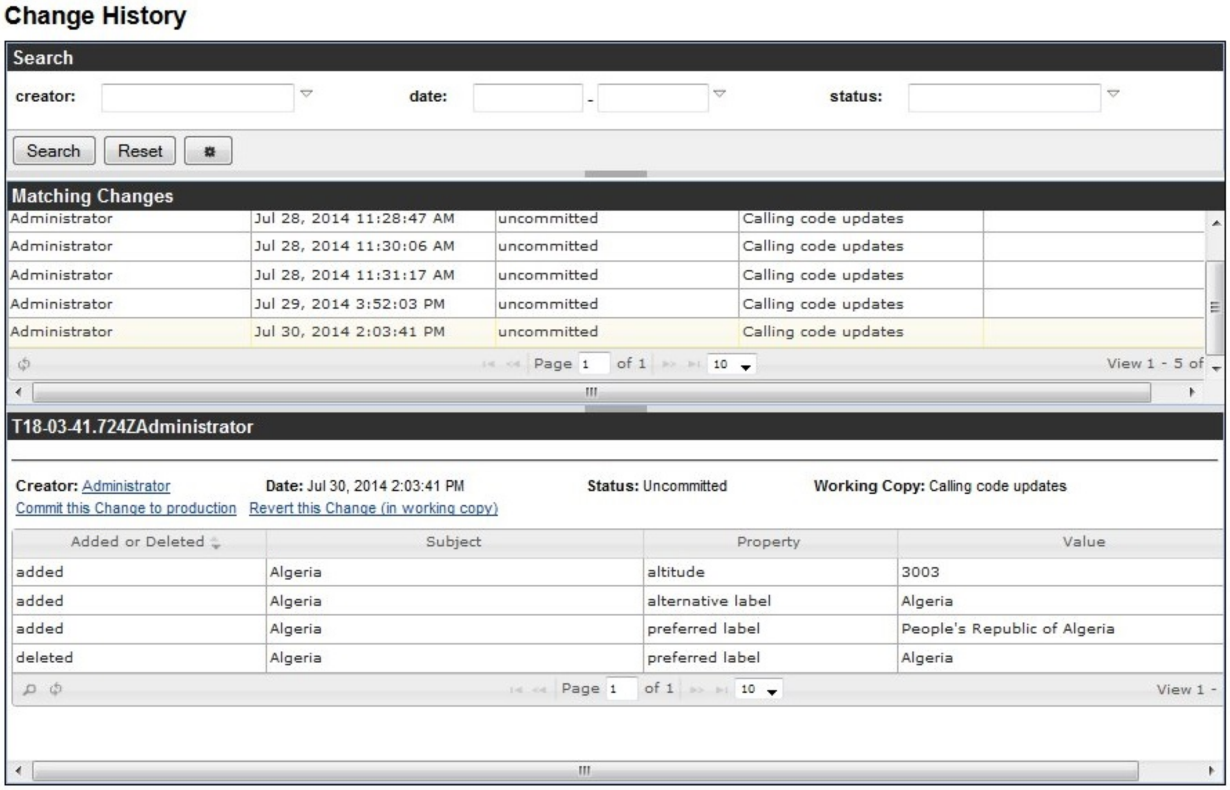
Validity checking also happens every time information about specific resource is update. In this case, validity is checked only for the updated resource. Users can also enable validity checking when they are viewing individual resources in the form. This setting is sticky, it will be remembered and applied across all of asset collections user works with. See the **View or Edit** documentation for details.

To develop custom extensions to this feature, see EDG Developer Guide > Extending the Problems and Suggestions Reports.

View Change History

Click Reports > View Change History to show the Change History view. For a production copy, this shows all the changes made since it has been created. For a workflow, this shows only changes made within the working copy managed by the workflow.

Clicking the Search button on the Change History screen displays a time-stamped list of the saves made in the Matching Changes panel, and clicking one of those lines displays details about what changes were made as part of that save operation in the Details of Selected Change panel. Below, the change made on July 30th has just been clicked, showing that three values were added and one was deleted as part of the change made with a particular save operation.



Change History

Search

creator: date: - status:

Matching Changes

Administrator	Jul 28, 2014 11:28:47 AM	uncommitted	Calling code updates	
Administrator	Jul 28, 2014 11:30:06 AM	uncommitted	Calling code updates	
Administrator	Jul 28, 2014 11:31:17 AM	uncommitted	Calling code updates	
Administrator	Jul 29, 2014 3:52:03 PM	uncommitted	Calling code updates	
Administrator	Jul 30, 2014 2:03:41 PM	uncommitted	Calling code updates	

T18-03-41.724ZAdministrator

Creator: Administrator **Date:** Jul 30, 2014 2:03:41 PM **Status:** Uncommitted **Working Copy:** Calling code updates

Commit this Change to production Revert this Change (in working copy)

Added or Deleted	Subject	Property	Value
added	Algeria	altitude	3003
added	Algeria	alternative label	Algeria
added	Algeria	preferred label	People's Republic of Algeria
deleted	Algeria	preferred label	Algeria

If you are logged in as a user who is editor or manager of the vocabulary/asset collection or a workflow where the change was made, then a link Revert this Change will appear in the bottom panel. Click on this link to undo this operation. This will in fact create a new "forward" edit in the change history, with yourself as author. Note that this feature should be used with care, because reverting some steps from the middle of the change history may lead to orphan resources in your model.

If you are logged in as a user who is editor or manager of an asset collection and look at a change performed in a working copy as part of a workflow, then a link Commit this Change to production will appear in the bottom panel. You can click on this link to move the change history entry (in the example above, the three additions and the deletion) out of the workflow copy and into the production copy, essentially cherry-picking which change from a workflow copy you want to accept. As with the Revert feature mentioned above, this feature should be used with care, because committing some steps from the middle of the change history may lead to creating data statements that are disconnected from the rest of the information. For example, when you commit a change that has modified some attribute of a newly created code, then you should also make sure that the change that created the code in the first place has also been committed.

Before you click the Search button, you can narrow the scope of the search by filling out any or all of the fields at the top of the form:

- **creator** Enter the name of a particular EDG user to only see changes by that user. This field uses typeahead, so that if you have users named "Joe" and "Joan" and only type in "Jo", these two names will appear in a drop-down list for you to pick from.

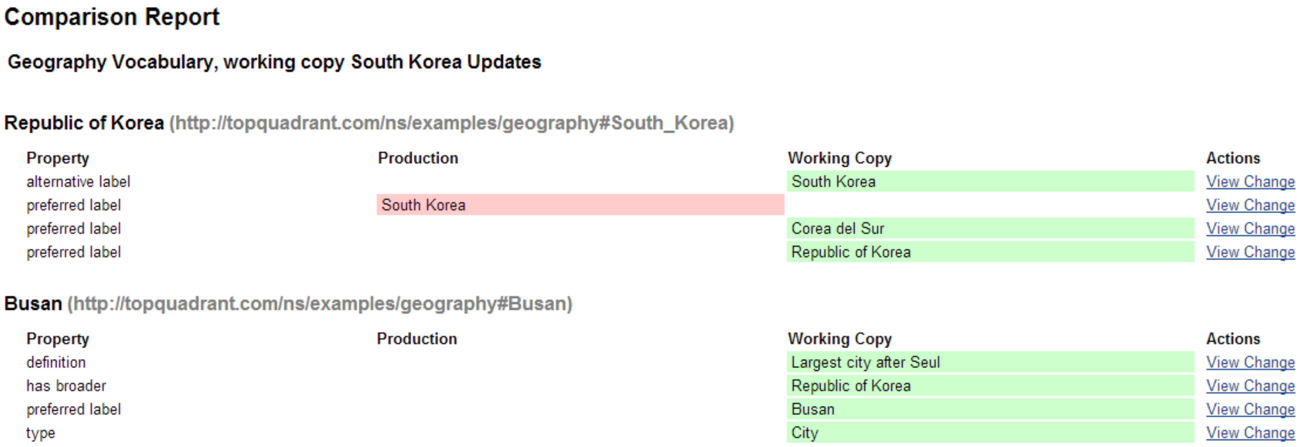
- **date** Enter a date in the first date field to see all changes after that date, a date in the second field to see all changes before that date, or in both fields to see the changes within a particular date range. (There's no need to actually type in the date value; clicking in either field displays a calendar where you can then click on the date you want to enter.)

- **status** Enter "committed" or "uncommitted" to only list changes with one of these status values.

Comparison Report

For a production Workflow, this report shows its differences with another, user-selected Workflow. For a working copy, it shows the differences to its parent production version. Note that differences do not extend to the contents of included asset collections. The report will list each changed assets and properties that were changed, showing the changed values. If a value was added, it is shown in green. If it was deleted, it is shown in pink.

For example, the following shows what happens after the preferred label property for "South Korea" is edited, an alternative label is added, and a "Seoul" is added as a narrower value of the "South Korea" resource (renamed to "Republic of Korea").



Comparison Report

Geography Vocabulary, working copy South Korea Updates

Republic of Korea (http://topquadrant.com/ns/examples/geography#South_Korea)

Property	Production	Working Copy	Actions
alternative label		South Korea	View Change
preferred label	South Korea		View Change
preferred label		Corea del Sur	View Change
preferred label		Republic of Korea	View Change

Busan (http://topquadrant.com/ns/examples/geography#Busan)

Property	Production	Working Copy	Actions
definition		Largest city after Seul	View Change
has broader		Republic of Korea	View Change
preferred label		Busan	View Change
type		City	View Change

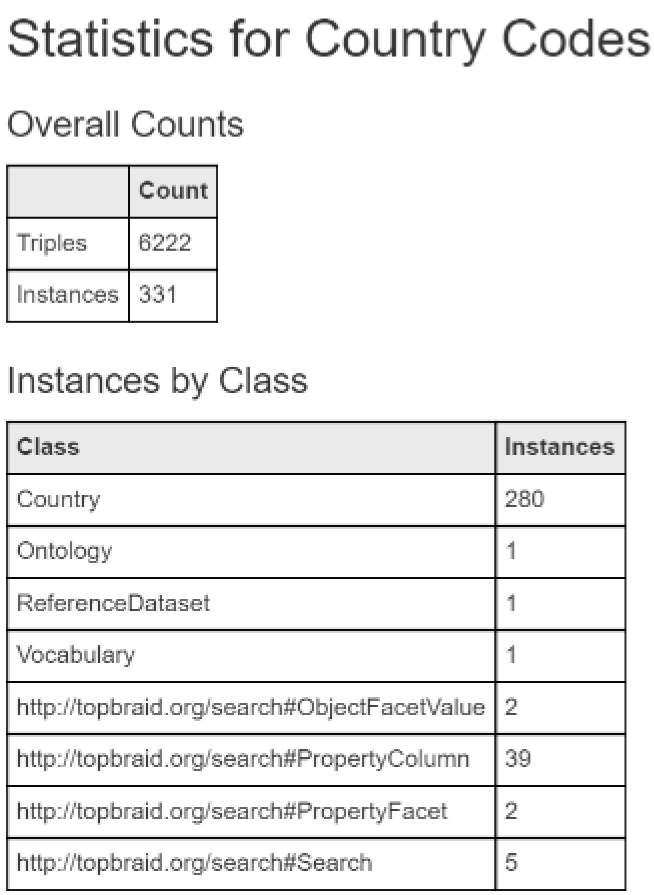
The right hand side of each change contains a link View Change that displays a dialog box with details of the change log entry that caused that particular change. Depending on your permissions, you can revert or commit the change in that dialog box. See View Change History for further information on reverting and committing individual changes.

Property Value Rules

This shows the collection's property values that are inferred through SHACL: `sh:defaultValue` or `sh:values` (see Inferring Data with SHACL Property Value Rules for details). The inferred properties are depicted diagrammatically.

Statistics

For any Workflow (production or working copy), Reports > Graph Statistics displays details about the Workflow's node distribution. The following shows the statistics for the sample Reference Dataset: *Country Codes*.



Statistics for Country Codes

Overall Counts

	Count
Triples	6222
Instances	331

Instances by Class

Class	Instances
Country	280
Ontology	1
ReferenceDataset	1
Vocabulary	1
http://topbraid.org/search#ObjectFacetValue	2
http://topbraid.org/search#PropertyColumn	39
http://topbraid.org/search#PropertyFacet	2
http://topbraid.org/search#Search	5

View Transition History

This lists the temporal sequence of the workflow's status transitions, executed by various users.

Comments

This lists all user comments associated with the workflow. Workflow-level comments can be created here directly. Asset-level (resource instance level) comments created while viewing or editing the workflow also appear here. A comment's status can be changed by any workflow editor (or manager) and by the comment's creator.