

# Terminology

## Page Contents

- 1 EDG Assets and Asset Collections
  - 1.1 Asset
  - 1.2 Asset collection
  - 1.3 Asset collection type (aka project type)
  - 1.4 Asset type
  - 1.5 Platform asset collection
- 2 EDG General
  - 2.1 Clone
  - 2.2 Dynamic Inferencing
  - 2.3 EDG enumeration collection
  - 2.4 EDG ontologies
  - 2.5 EDG operational governance model
  - 2.6 Event
  - 2.7 Event type
  - 2.8 Impact
  - 2.9 Includes
  - 2.10 Lineage
- 3 EDG Modeling
  - 3.1 Attribute (aka datatype property)
  - 3.2 Class
  - 3.3 Constraint (aka SHACL constraint)
  - 3.4 Instance (aka class instance, class member, individual)
  - 3.5 Node shape
  - 3.6 Ontology
  - 3.7 Property
  - 3.8 Property shape
  - 3.9 Range (of values)
  - 3.10 Relationship (aka object property)
  - 3.11 Resource
  - 3.12 Shape (aka SHACL shape)
- 4 EDG Taxonomy
  - 4.1 Concept
  - 4.2 Scheme
  - 4.3 Taxonomy
- 5 EDG Teamwork
  - 5.1 Comment
  - 5.2 Teamwork permission profiles: viewer, editor, manager
  - 5.3 Production copy
  - 5.4 Task
  - 5.5 Workflow
  - 5.6 Workflow template
  - 5.7 Working copy

## EDG Assets and Asset Collections

### Asset

An asset is a technical, business, or operational resource governed by an organization using TopBraid EDG. Examples of assets could include databases, business applications, vocabulary terms, reference data, requirements, and other technical or enterprise resources.

### Asset collection

Assets are organized into collections, which are stored technically as named graphs. You can think of collections as datasets. Each asset collection must have at least one manager as well as, optionally, any number of users with the edit and view privileges. See [Users and Access Control](#) to understand EDG permissions system.

In addition to permissions, Asset collections have a variety of other metadata such as description, subject area they belong to, etc. This metadata can be viewed and edited by going to the Settings tab of an asset collection.

Collections can include (*cf. includes*) each other by reference. When editing information in one collection, users can see and link to any information in the included collection. They cannot, however, delete or change any information stored in the included collection. Most asset collections are based on some ontology that defines schema for the data (assets) they hold. (*cf. EDG ontologies*)

## Asset collection type (aka project type)

Each asset collection has exactly one type. It determines what kind of assets are stored in the asset collection, what kind of metadata (attributes and relationships) is captured about an asset collection and what functionality such as imports, exports, reports, editing applications, etc., are available for it. TopBraid EDG includes many project types such as Glossary, Reference Dataset, Lineage Model, Ontology, etc. For a full list of asset collection types with their descriptions see [this page](#). Users can also create their own asset collection (project) types.

Collection types available in a given installation of EDG are determined by the license and, optionally, any additional exclusions made by the EDG administrator. You will see all collection types available in your installation in the Navigation Bar on the left hand side of EDG pages.

## Asset type

An asset can have one or more types. Asset type is a class in an ontology (cf. *class*, *ontology*), either ontologies shipped with TopBraid EDG or customized/created by the users, that formally describes attributes and relationships of an asset. TopBraid EDG includes over 100 asset types such as Glossary Term, Database, Requirement, etc. Since asset type is a class in an ontology, depending on your perspective, it may also be an asset.

## Platform asset collection

When users start using EDG, they create their own asset collections to describe assets they want to govern with EDG. Platform asset collections are already pre-created when EDG is installed. There are two platform asset collections - EDG Enumerations (cf. *EDG enumerations collection*) and EDG Governance Model (cf. *EDG operational governance model*).

## EDG General

### Clone

A copy of a resource with all its metadata. Creating and editing clones can be a useful way to create new resources with minimal data entry.

### Dynamic Inferencing

A process by which EDG calculates "just in time" property values for properties that contain a rule as part of their property shape definition. (cf. property *shape*). If a property value is inferred, it is not editable in EDG, unless a rule only specifies a default value. In a latter case, the property value remains editable in EDG so that a user could replace a default.

### EDG enumeration collection

An EDG Platform Asset Collection (cf. *platform asset collection*) that stores controlled lists of values used across all asset collections e.g., a value list for statuses. EDG Administrators can set up these lists by going to **Server Administration > EDG Configuration Parameters > Setup EDG Enumerations**. For user convenience, TopBraid EDG ships with some pre-built value lists. These are organized into codelists stored as files in the EDG workspace. As par of enumerations setup, administrators can load the pre-built codes from EDG codelists or create their own preferred values.

### EDG ontologies

Pre-built ontology models (cf. *ontology*) shipped with EDG that define over 100 asset types (cf. *asset type*) relevant to data governance. There is a model corresponding to nearly every asset collection type (cf. *asset collection type*). These models are stored as files in EDG workspace. They are customizable. To customize one of EDG ontologies, create an ontology in EDG; include (cf. *includes*) the model you want to customize; make changes and extensions. See [Getting Started with Business Glossaries](#) for an example of how this process works.

In the Includes dialog, EDG ontologies show up with the name "EDG Shapes - <asset collection name>".

### EDG operational governance model

An EDG Platform Asset Collection (cf. *platform asset collection*) that stores information about organizational structure, roles, responsibilities and process for data governance. Any user with permissions to see this asset collection can access it by clicking on one of the links under the Governance Model heading in the left hand-side Navigation Bar on EDG pages.

### Event

A change in data stored in TopBraid EDG that EDG is watching for and is prepared to act upon.

### Event type

Each event has exactly one Event Type. Event Type formally describes what data change indicates that event has occurred and what action TopBraid EDG should take when it happens e.g., send a notification e-mail. TopBraid EDG pre-defines several event types. For example, a change in a working copy status. Users can create additional event types.

## Impact

Impact is a reverse of Lineage (cf. *lineage*). It shows the flow of data from a data element or a dataset user is focused on to its destination in other assets in the enterprise. Just as lineage, impact information is presented in an interactive diagram. To access it, click on the asset of interest and then click on the Visualization Actions  menu icon and pick Impact from the available options.

## Includes

Asset collections can be included into each other.

When collection A includes collection B (or file B), users working with A get access to all assets in B. However, information stored in collection B can not be modified - access to them is view-only. In some cases, icons displayed in the EDG UI for included resources are shaded to indicate that they are included. For example, when ontology A includes ontology B, classes from B are displayed with shaded icons. When searching within an asset collection, users can limit search to only "local assets" i.e., those that do not come from the included collections.

Inclusion is accomplished by going into **Settings > Includes** from a collection's home page. EDG has some (configurable) rules about the types of collections that could be included into each other. Includes dialog also lets user include files that are either in RDF format or can be auto-converted by EDG on the fly e.g., spreadsheet.

## Lineage

Data lineage of a particular data element or a dataset identifies the data's origins and what happens to it as it goes through diverse processes from its origin to the element/dataset of interest. Data lineage can be captured and presented by EDG at the lowest level of data flows details - actual tables, scripts and statements. It can also be captured and presented at the higher, business level, connected to business terms and processes. And it can be rolled up and drilled down as necessarily for the different stakeholders and use cases.

The simplest form of lineage information can be captured in the Data Asset Collections using "maps to" relationship. A more comprehensive treatment of lineage is supported by the Lineage Model asset collections. Lineage information is presented in an interactive diagram called

LineageGram. To access it, click on the asset of interest and then click on the Visualization Actions  menu icon and pick Lineage from the available options. Or, if your starting point is *Search the EDG* search results, LineageGram icon is shown next to each search result that has lineage information.

## EDG Modeling

### Attribute (aka datatype property)

An attribute is a specific piece of information that you capture for an asset, such as a name or a short textual description. Each attribute has a range of values of some literal type (e.g., text, numbers, etc.) (cf. *property, relationship*).

### Class

A resource that identifies and formally describes a set of resources that have some property characteristics in common e.g., class of databases, database tables, organizations, etc. The description is typically done in terms of possible properties and property values e.g., an organization may have a contact e-mail address, a database table belongs (is a table of) some database, etc. Resources described by a class are called class members or class instances. A class can also be a shape (cf. *shape*).

### Constraint (aka SHACL constraint)

Part of a shape (cf. *shape*) that constraints what values are valid for a specific property of a given set of resource e.g., min and number of values, their type, their relationship with other values.

### Instance (aka class instance, class member, individual)

Typically, these terms are used to refer to resources that are not part of the data schema e.g., are not classes or properties. While classes, properties and shapes define data schema, instances are data.

### Node shape

A shape (cf. *shape*) that describes information about target resources themselves (e.g., shape of the URI) and groups together all applicable property shapes.

## Ontology

A description of entities in some area of interest, captured using RDFS, OWL or SHACL. An ontology is an information model and an asset collection type in EDG. It contains schema elements - classes, properties, shapes and rules. May also contain some instances.

## Property

An attribute or relationship associated with a given class (cf. *attribute, relationship*).

## Property shape

A shape (cf. *shape*) that describes information about values of a specific property. A property shape contains one or more constraints. It can also contain "non constraining" information e.g., display name or calculation/inferencing rule for property values.

## Range (of values)

A range defines what values are possible for a specific attribute or a specific relationship. Ranges for attributes are mainly standard XML datatypes such as string, integer and date. HTML datatype is also supported for storing rich text. Ranges for relationships are classes. For example, in case of the "column of" relationship between a Database Column and a Database Table, the range of relationship is the class Database Table. Range of values for attributes is typically specified using "datatype" constraint. Range of values for relationships is typically specified using "class" constraint.

## Relationship (aka object property)

This is a directional link between exactly two resources. It captures how they are related to each other. Each relationship has a range of values (cf. *property, attribute*).

## Resource

This is anything you want to capture information about using TopBraid EDG. Asset is a resource. Asset collection is a resource. Properties (attributes and relationships) are resources, etc. Each resource has a globally unique URI. Formally speaking, a resource is any object that is uniquely identifiable by a URI, a *uniform resource identifier*. It is used by the web infrastructure you are familiar with. URLs are URIs, as are e-mail addresses.

## Shape (aka SHACL shape)

Shape describes characteristics of target resources e.g., what property (cf. *property*) values they may have, how their URIs may look like, etc. Target of a shape may be defined as all members of some class (cf. *class*) or as an individual resource (cf. *instance*) or as all resources that have any value for some specified property, etc.

In EDG, classes are typically also shapes which provide a complete definition of all class properties. Additional shapes targeting a class are defined to provide alternative, role-specific views on data. For example, generally speaking, organizations can have descriptions, addresses, phone numbers, sub-organizations, organization members, etc. Organization class/shape will describe all these properties. An alternative shape may only include name, description and web address property - to provide an abridged view into the available information about an organization.

## EDG Taxonomy

### Concept

The basic asset in EDG Taxonomy. A concept is usually known by its preferred label, and can have various kinds of metadata assigned to it.

### Scheme

This is a set of concepts grouped together into a list or hierarchy. It might represent a taxonomy, a thesaurus, a code list, or any other controlled vocabulary. A vocabulary may be a single scheme, but because of EVN's ability to group several vocabularies together, some may appear as multiple schemes. For example, you might have a taxonomy of apparel products and another of colors in which the clothing was available both displayed at the same time.

### Taxonomy

An asset collection storing a set of business concepts described using SKOS or SKOS-XL. These are typically used for taxonomies, vocabularies, or subject headings that are hierarchical in nature.

## EDG Teamwork

The teamwork system maintains workflows and associated working copies of asset collections, along with change history, comments and tasks.

## Comment

An object that can be created by a user to capture input, question or an issue with an asset, an asset collection or a task. Even if a user doesn't have edit privileges for an asset collection, they can still create comments about assets described in the collection. Comments can have statuses.

## Teamwork permission profiles: *viewer, editor, manager*

*Teamwork* is an EDG framework that controls the access and life-cycles of its asset collections. The three Teamwork permission profiles: *viewer*, *editor*, and *manager*, provide nested levels of collection and asset functions to users (assigned as individuals or as *security roles*). For each asset collection or working copy, a user's access is determined by the permission profile (V/E./M) assigned to them or their security role(s). For example, users will not see any asset collection for which they lack at least a *viewer* level permission. Editors (including managers) are able to create and modify the assets in a collection. Only managers will see as collection's *Manage* view or be able to change permission profiles of other users.

## Production copy

The official version of an asset collection that is currently in use (cf. *working copy*).

## Task

An object that is created to capture a work item associated with an asset or an asset collection. A task has to have an assignee and a status. It may have a due date. It may also have comments.

## Workflow

A process for making changes to asset information in a sandboxed environment (cf. *working copy*), taking these changes through any necessary review, approval and disposition. Workflow is defined as a set of states with actions and roles that can modify a state.

## Workflow template

A workflow template defines a workflow, what asset collections it can be used with, what states it can go through, etc. EDG is shipped with a pre-built "default" workflow template. Users can create additional templates.

## Working copy

This is a branched copy of a production asset collection, which isolates its editing, review, and approval activities. A working copy may go through a workflow approval process, after which its changes may or may not be committed back into the official production version. There may be multiple simultaneous working-copy instances as users in various profiles make and review changes in parallel.