

# Qudex Object Model

## Version Information

Previous version: 3.0c

This version: 3.0d

Date: 29/01/13

## Changes from 3.0c to 3.0d

Added two new attributes to the “CategoryType” model object. These attributes are CategoryScheme and CategoryType, which are described in the QuDEX reference model document (DExT\_quali-QuDEX\_v03\_01\_reference\_01\_01.doc).

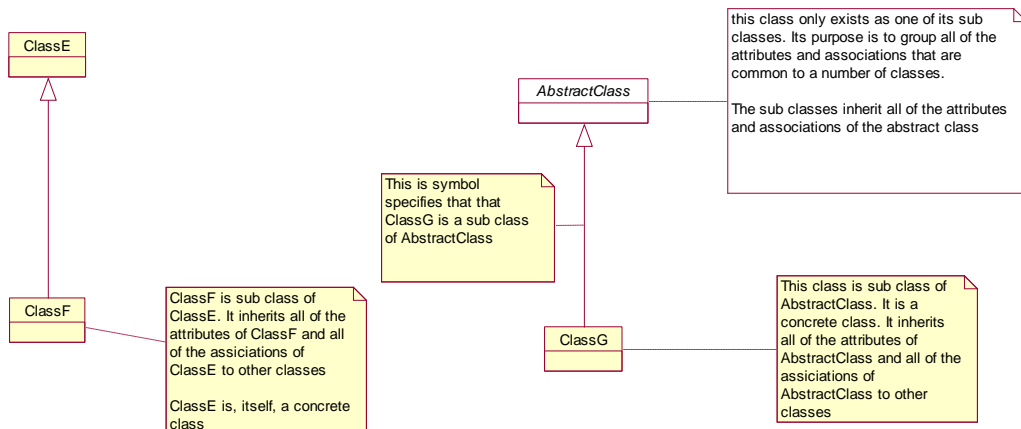
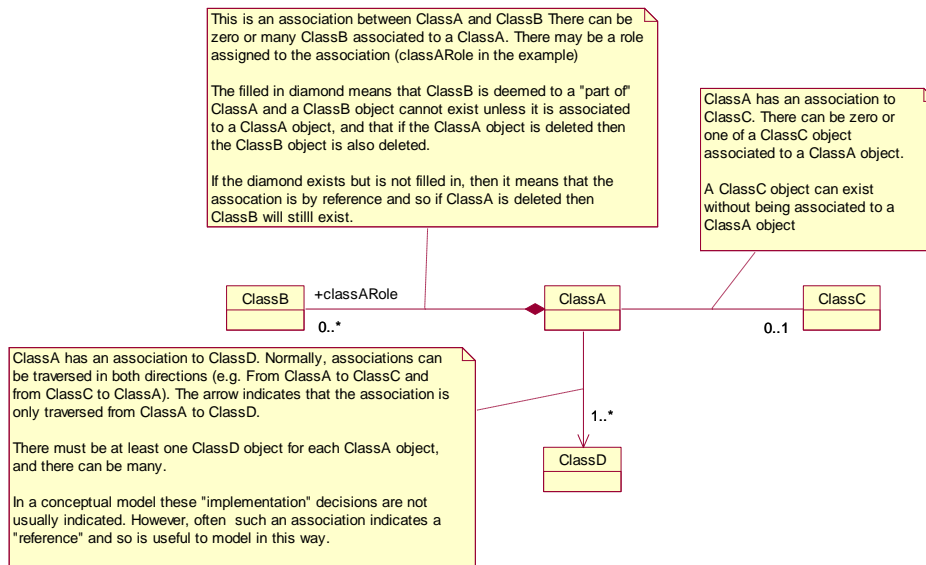
## Changes from 3.0b to 3.0c

Corrected some typos in the UML modelling notation diagram.

## Changes from 3.0 to 3.0b

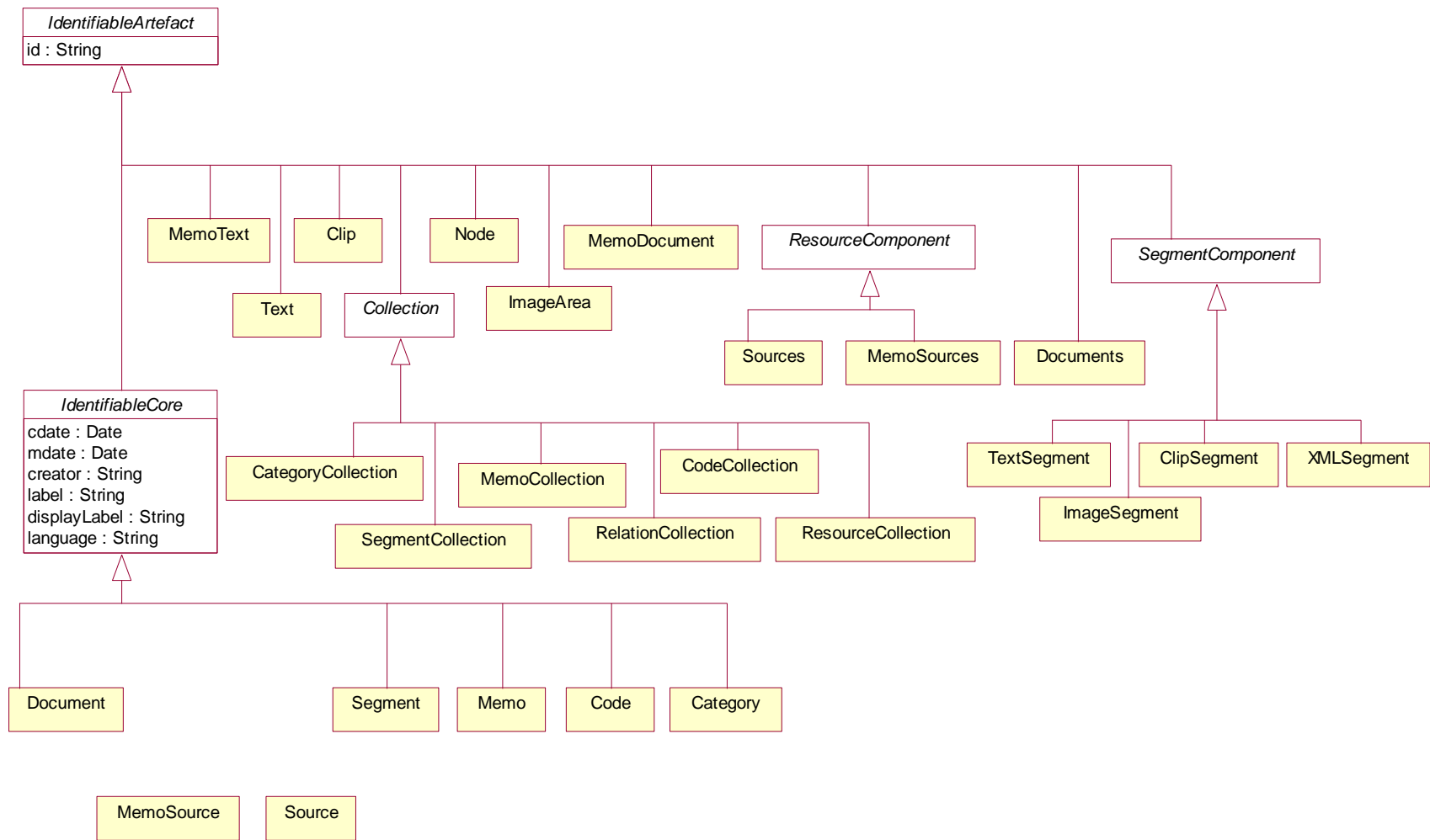
- 1) Re-modelled the ResourceComponent, Sources, MemoSources and Documents.
- 2) Changed the relationship between Document and SourceResource (this has been deleted in (1) above) to Resource Component” and added documentType attribute with DocumentType enumeration of "source" and "memoSource".
- 3) Changed cardinality between ResourceCollection and ResourceComponent from 1..\* to 0..\* (this reflects the schema)
- 4) Changed cardinality between Documents and Document from 1..\* to 0..\* (this reflects the schema)
- 5) Changed name of XMLRange to Node

## UML Modelling Notation



## Identity and Inheritance.

**Model Diagram**



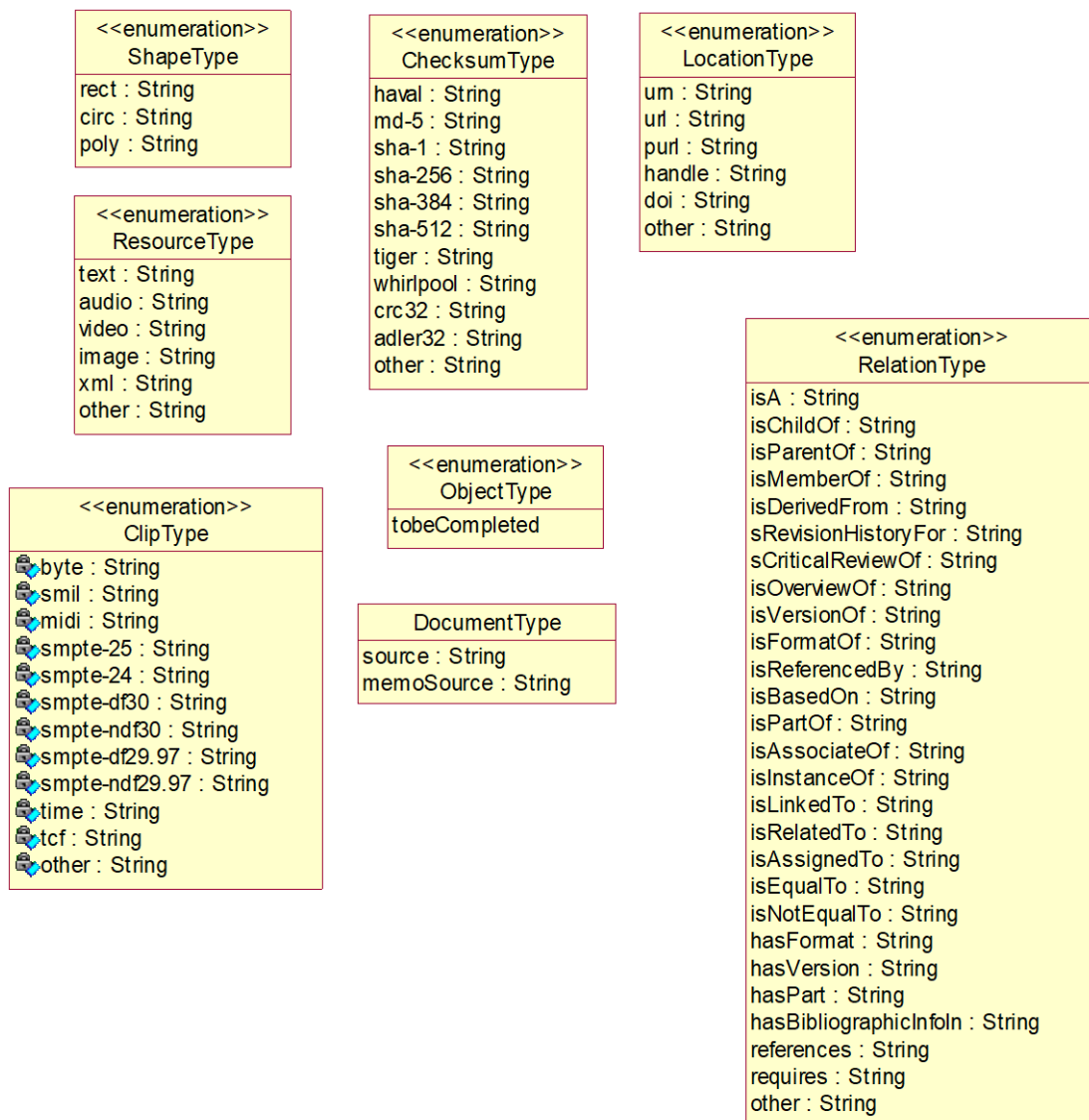
## Diagram Explanation

There are many classes which have an Id attribute. This is important as this Id is used to relate two objects together in the Relation Collection – this is shown later.

Some classes also have common attributes in addition to the identity. These inherit from the Identifiable Core, which itself inherits from Identifiable Artefact.

## Data Types

### Model Diagram

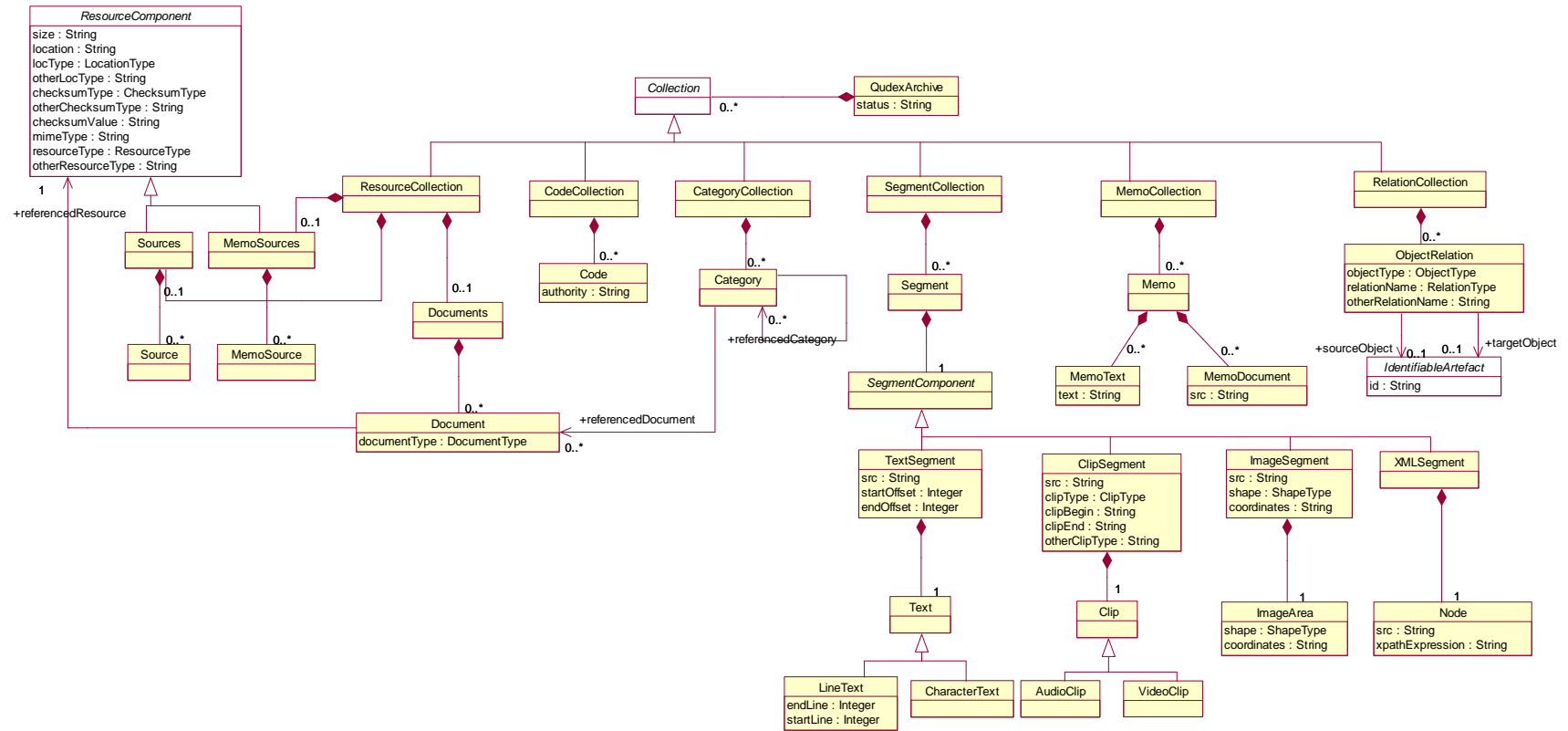


## **Diagram Explanation**

These data types are used to constrain the possible values of attributes in some of the Qudex classes – see later.

# Qudex Relationships

## Model Diagram



## **Diagram Explanation**

This model is an object oriented representation of the Qudex schema. Where appropriate the names of attributes and elements are taken from the equivalent construct in the schema.

One objective of this model is to discover commonalities between the objects in Qudex and to ensure these can share common properties. Therefore, in many cases two or more classes are sub classes of a class that contains the common components.