# Plasma Data Studio

TerraMeta Software, Inc.

PlasmaSDO® and PlasmaQuery® are registered
Trademarks of TerraMeta Software, Inc.

# 1 Introduction

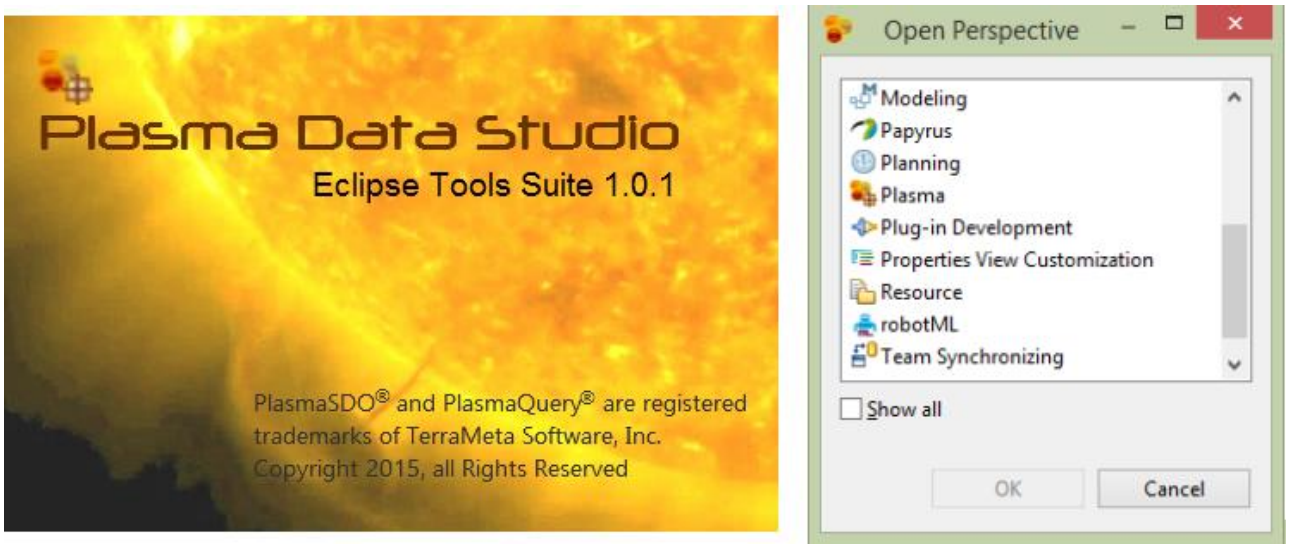
Plasma is a data store agnostic, object mapping and object query framework written in a Java™ with Maven tools for metadata ingestion and conversion. At its core, Plasma contains a directed graph or digraph model and a set of metadata driven graph traversal algorithms. Data Objects under Plasma form a digraph transparently as a client manipulates the SDO API, graph edges or links being automatically created and used internally to manage associations between Data Object nodes. Data store provider implementations are available for Apache HBase, MAPR-DB (M7) and various relational database vendors.

The Plasma metadata extensions are described in a UML profile which contains UML stereotypes, enumerations and other elements used to enrich UML models for use within the Plasma core as well as third party Data Access Service (DAS) providers. Particular design consideration has been focused on leaving each stereotype granular with only a few tightly related attributes/tags, rather than more monolithic stereotype groupings. This approach lets each stereotype convey far more meaning and maps well to metadata oriented extensions in various target languages, such as Java™ annotations. This granular approach can however have the effect of making UML diagrams more cluttered depending on the presentation settings of the UML diagramming tool.

A single UML logical model fully enriched or annotated with the Plasma UML profile provides enough context specific information to support various technology-specific runtime environments and the generation of numerous context or platform-specific models as well as many other related source-code and other artifacts.

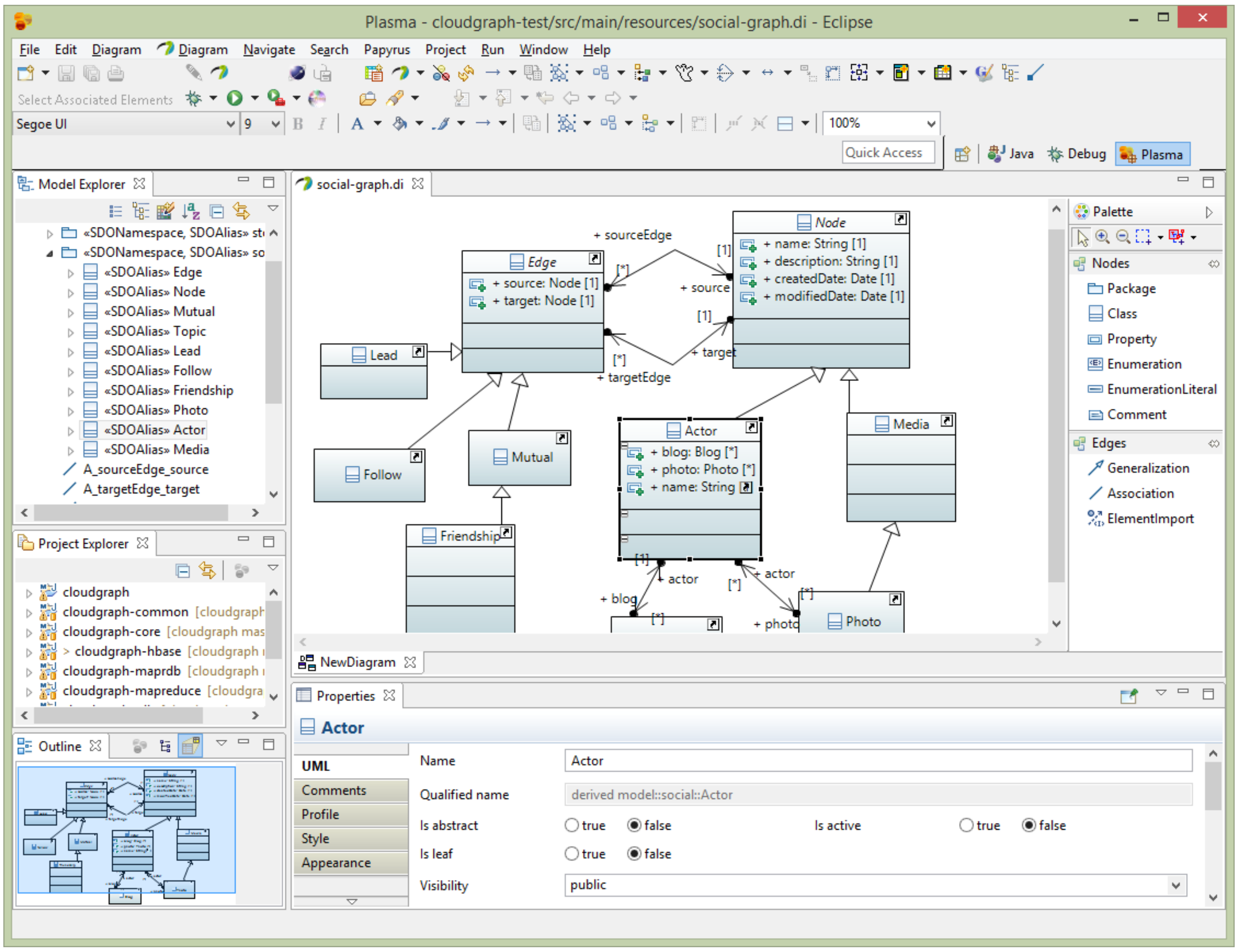
# 2 Plasma Data Studio

Plasma Data Studio is an Eclipse based IDE for annotating and integrating UML models with Maven (M2E) based Java projects. The UML editor is derived from Eclipse Papyrus components. See https://www.eclipse.org/papyrus. It is centered around a custom Eclipse perspective (Plasma perspective), capability to create projects and diagrams, and a custom UML palette. UML Models are annotated using a custom UML profile and datatypes package, both of which are fully integrated or registered in the IDE. The below screen shot pictures a UML model being edited in-line with a standard Java Eclipse project.



PlasmaSDO® and PlasmaQuery® are registered trademarks of TerraMeta Software, Inc. Copyright 2015, all Rights Reserved

**Figure 1 - Plasma Perspective Dialog**



Actor

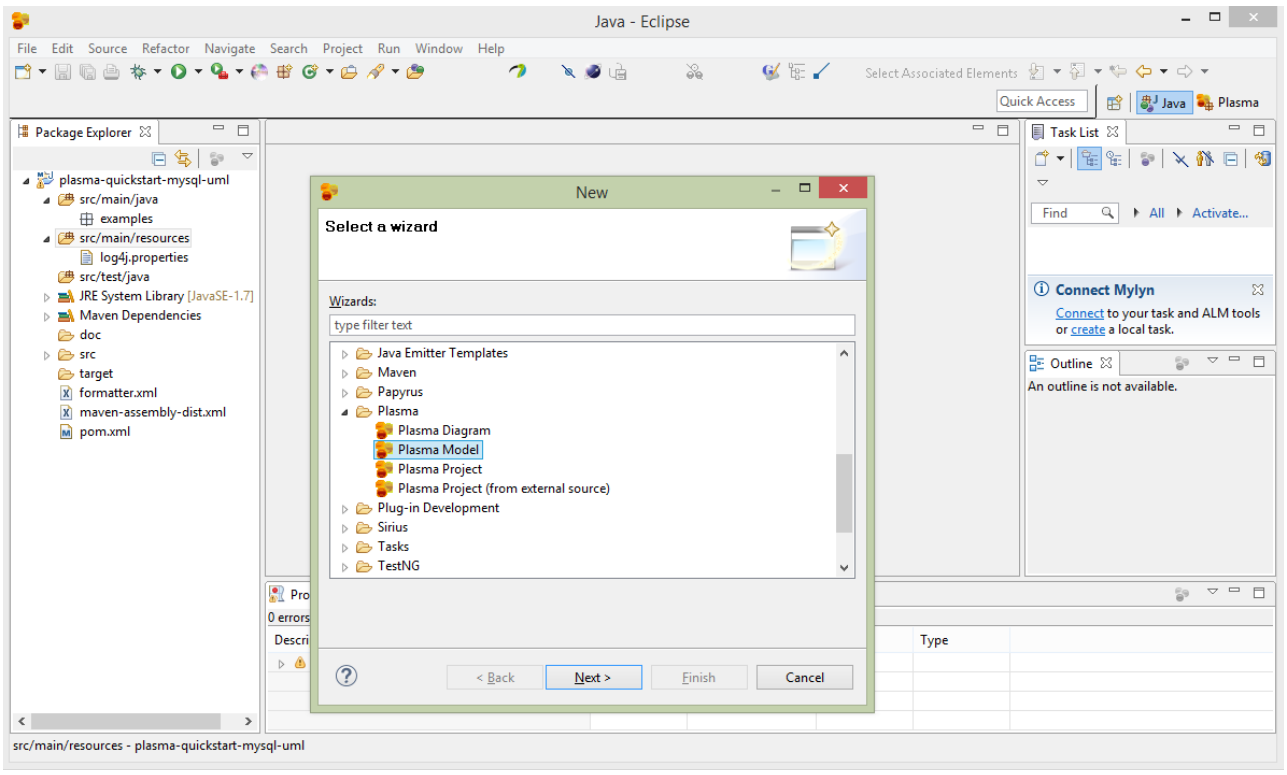
| | | | |
|---|---|---|---|
| Name | Actor | | |
| Qualified name | derived model::social::Actor | | |
| Is abstract | true [✓ false] | Is active | true [✓ false] |
| Is leaf | true [✓ false] | | |
| Visibility | public | | |

**Figure 2 - Plasma Data Studio**

## 3 Creating a Plasma Model

From Plasma Data Studio create a new UML model and diagram into the src/main/resources directory of your existing maven project.

1. Right click on the resources directory, then to New->Other->Plasma as below.
2. Save the model (it will have a .di extension, for diagram) in the resources dir of the project.
3. Select Plasma Class Diagram for the diagram type. Then Finish the wizard.



Package Explorer:

- plasma-quickstart-mysql-uml
  - src/main/java
    - examples
  - src/main/resources
    - log4j.properties
  - src/test/java
  - JRE System Library [JavaSE-1.7]
  - Maven Dependencies
  - doc
  - src
  - target
  - formatter.xml
  - maven-assembly-dist.xml
  - pom.xml

New

**Select a wizard**

Wizards:

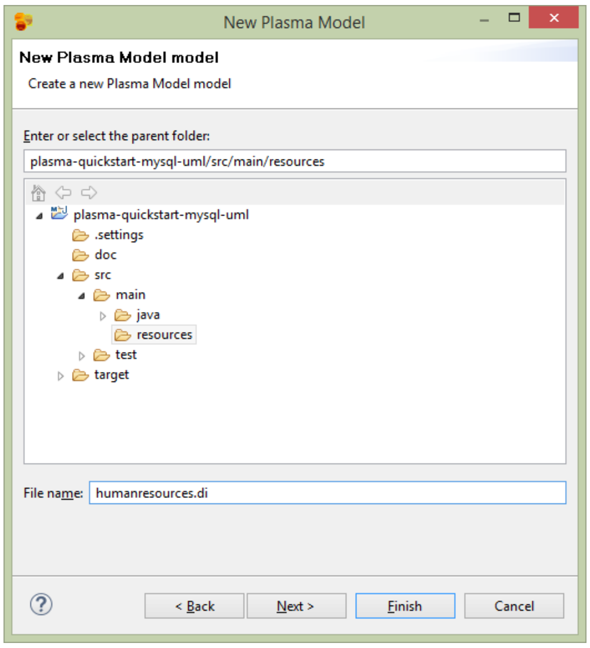
type filter text

- Java Emitter Templates
- Maven
- Papyrus
- Plasma
  - Plasma Diagram
  - Plasma Model
  - Plasma Project
  - Plasma Project (from external source)
- Plug-in Development
- Sirius
- Tasks
- TestNG

Connect to your task and ALM tools or create a local task.

An outline is not available.

src/main/resources - plasma-quickstart-mysql-uml



**New Plasma Model model**

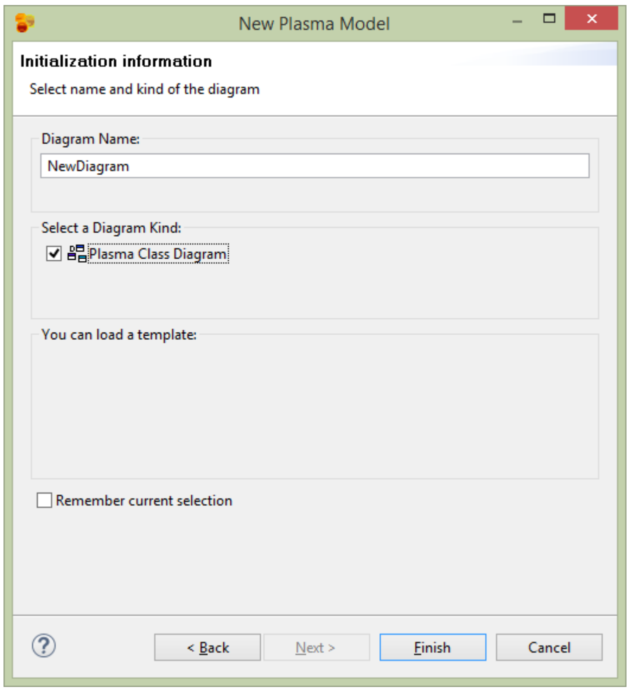
Create a new Plasma Model model

Enter or select the parent folder:

plasma-quickstart-mysql-uml/src/main/resources

- plasma-quickstart-mysql-uml
  - .settings
  - doc
  - src
    - main
      - java
      - resources
    - test
  - target

File name: humanresources.di


New Plasma Model

Initialization information

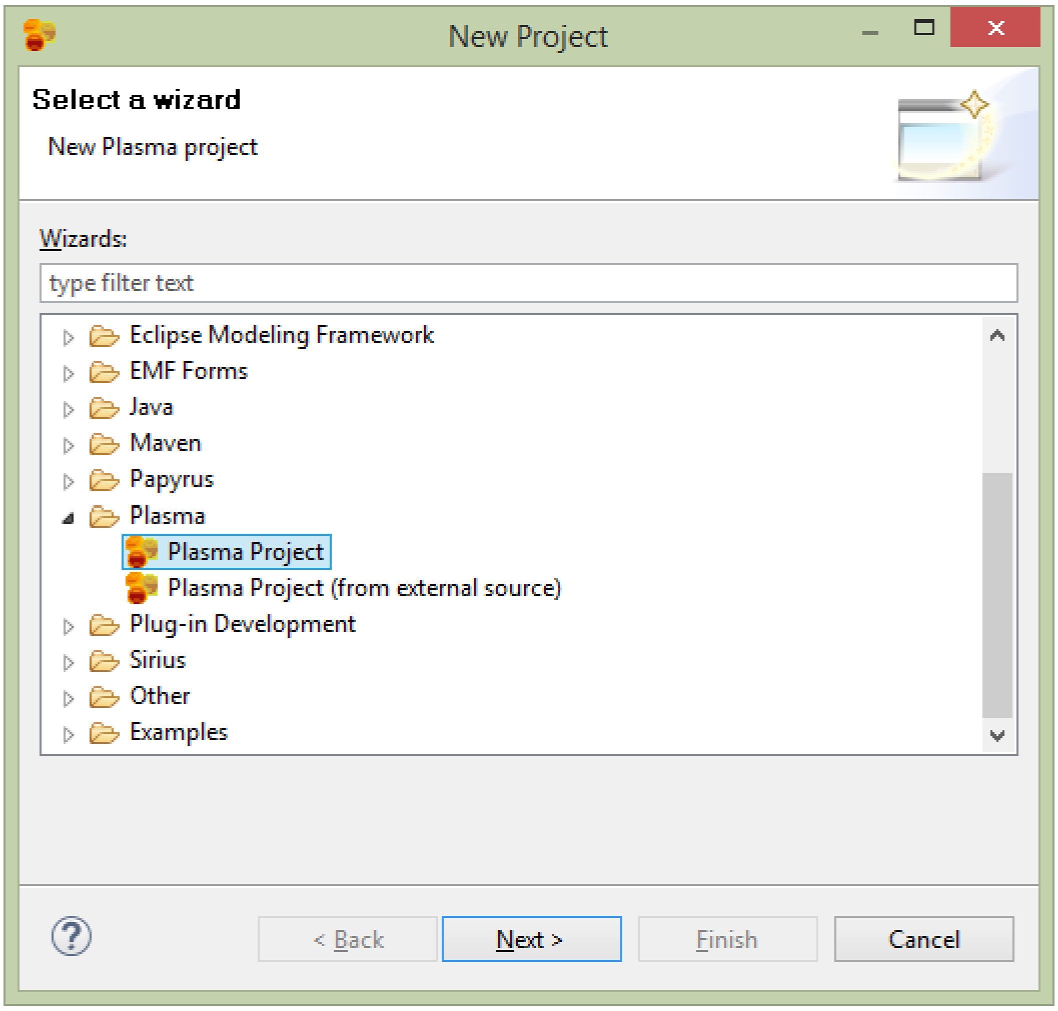
Select name and kind of the diagram

Diagram Name: NewDiagram

Select a Diagram Kind: [✓] Plasma Class Diagram

You can load a template:

[ ] Remember current selection


New Project

Select a wizard

New Plasma project

Wizards:

type filter text

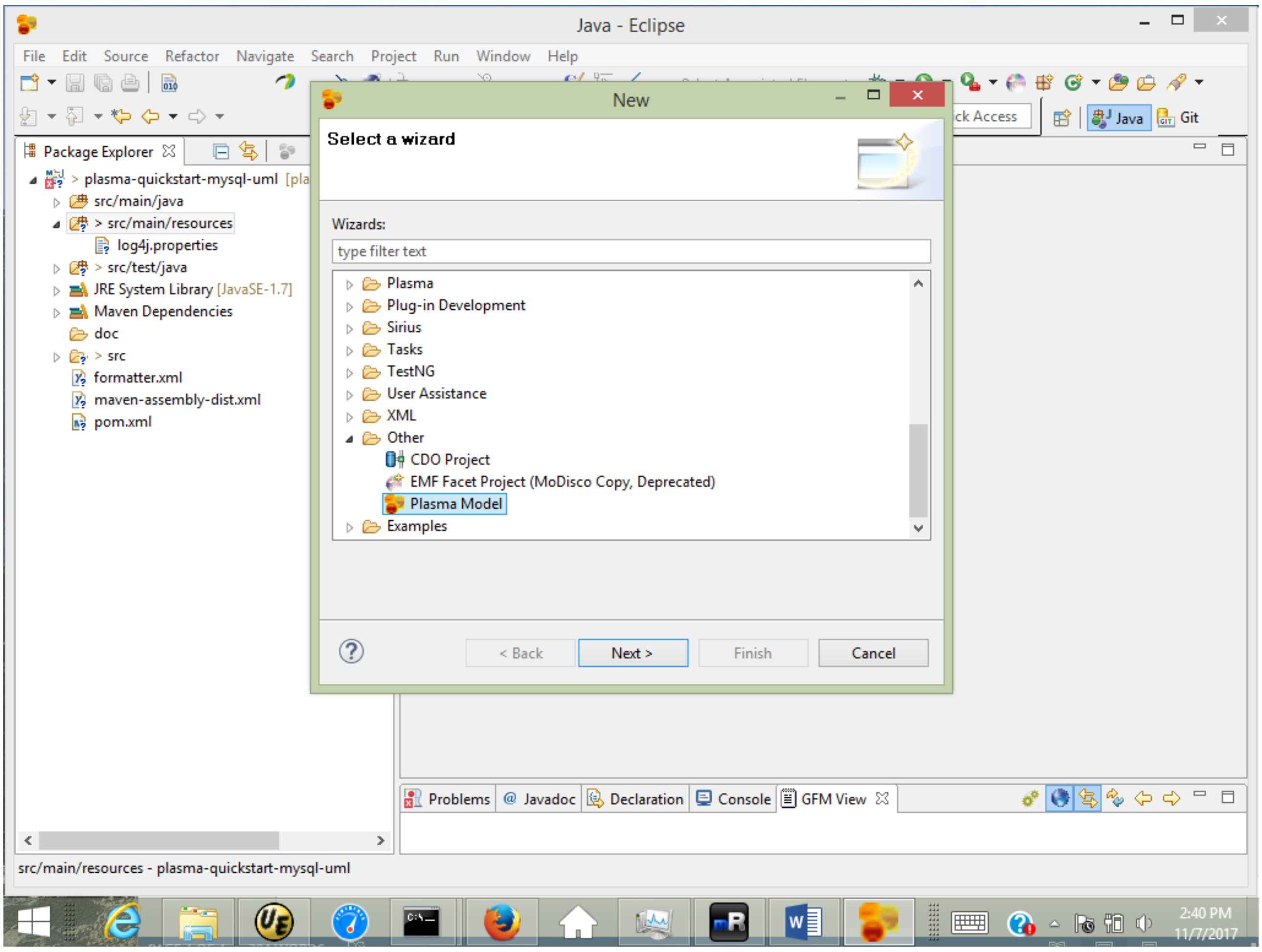
- Eclipse Modeling Framework
- EMF Forms
- Java
- Maven
- Papyrus
- Plasma
  - Plasma Project
  - Plasma Project (from external source)
- Plug-in Development
- Sirius
- Other
- Examples

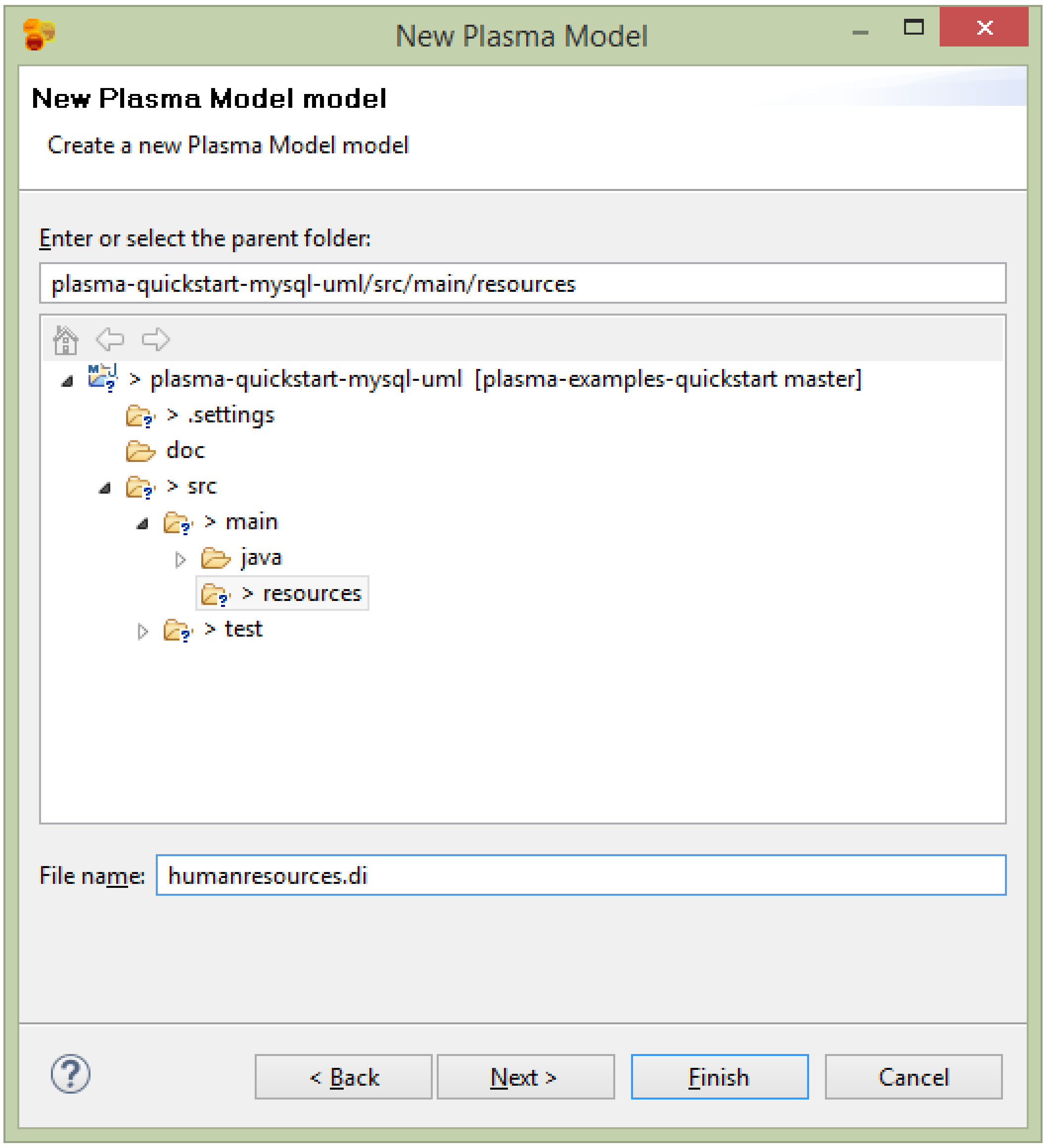
**Figure 3 - Create Project Wizard**

# 4 Creating a Plasma UML Model

Create a new UML model and diagram into the src/main/resources directory of your existing maven project.

1. Right click on the resources directory, then to New->Other->Plasma as below.
2. Save the model (it will have a .di extension, for diagram) in the resources dir of the project.
3. Select Plasma Class Diagram for the diagram type. Then Finish the wizard.





**New Plasma Model model**

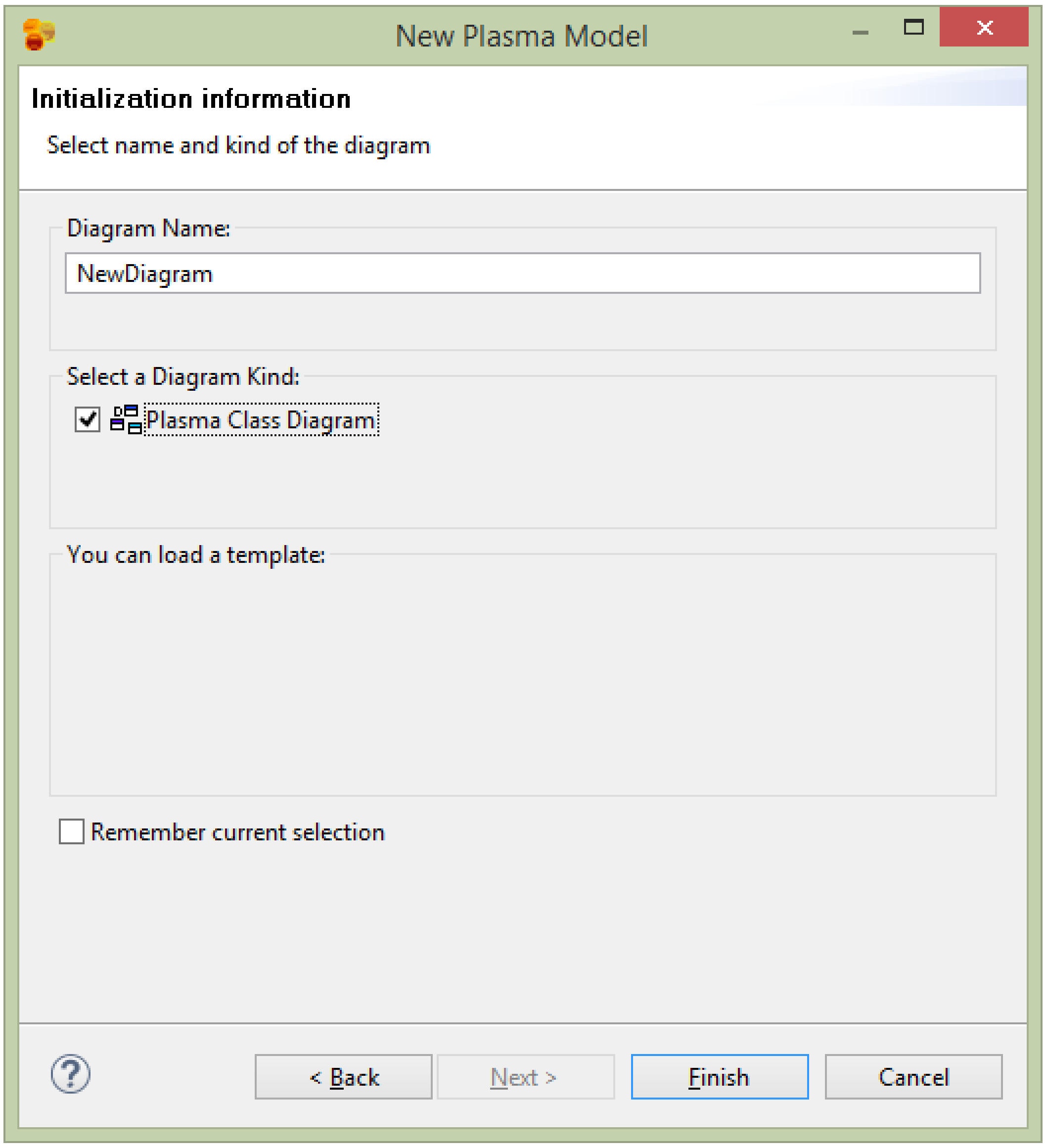
Create a new Plasma Model model

Enter or select the parent folder:

plasma-quickstart-mysql-uml/src/main/resources

```
> plasma-quickstart-mysql-uml  [plasma-examples-quickstart master]
    > .settings
    doc
    > src
        > main
            java
            > resources
        > test
```

File name: humanresources.di


New Plasma Model

**Initialization information**

Select name and kind of the diagram

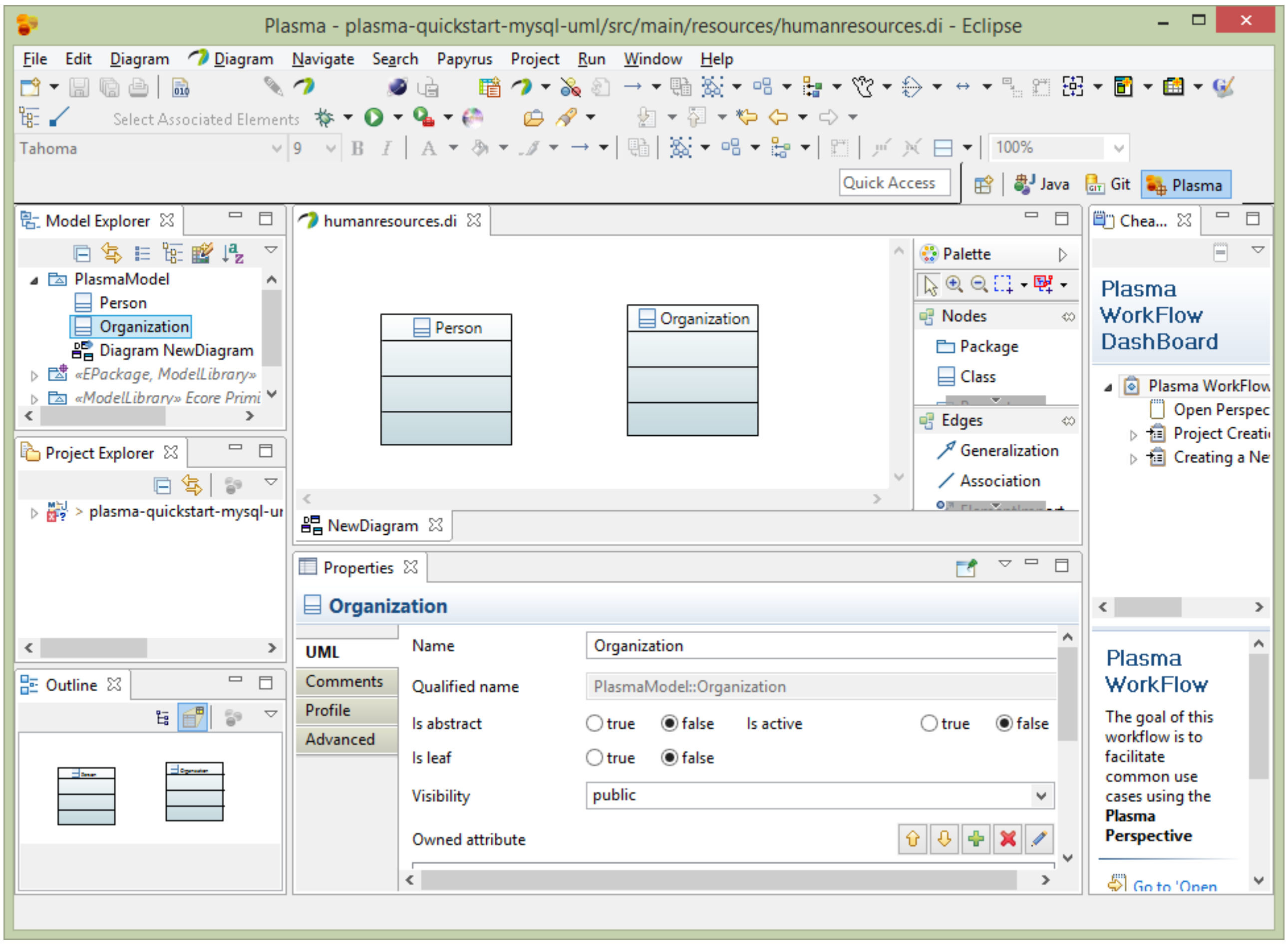
Diagram Name: NewDiagram

Select a Diagram Kind:

[✓] Plasma Class Diagram

You can load a template:

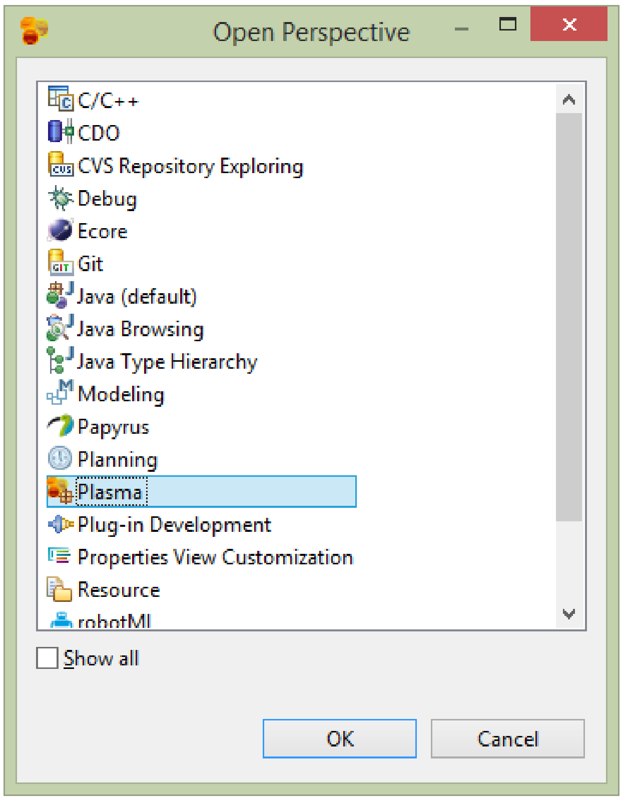
[ ] Remember current selection

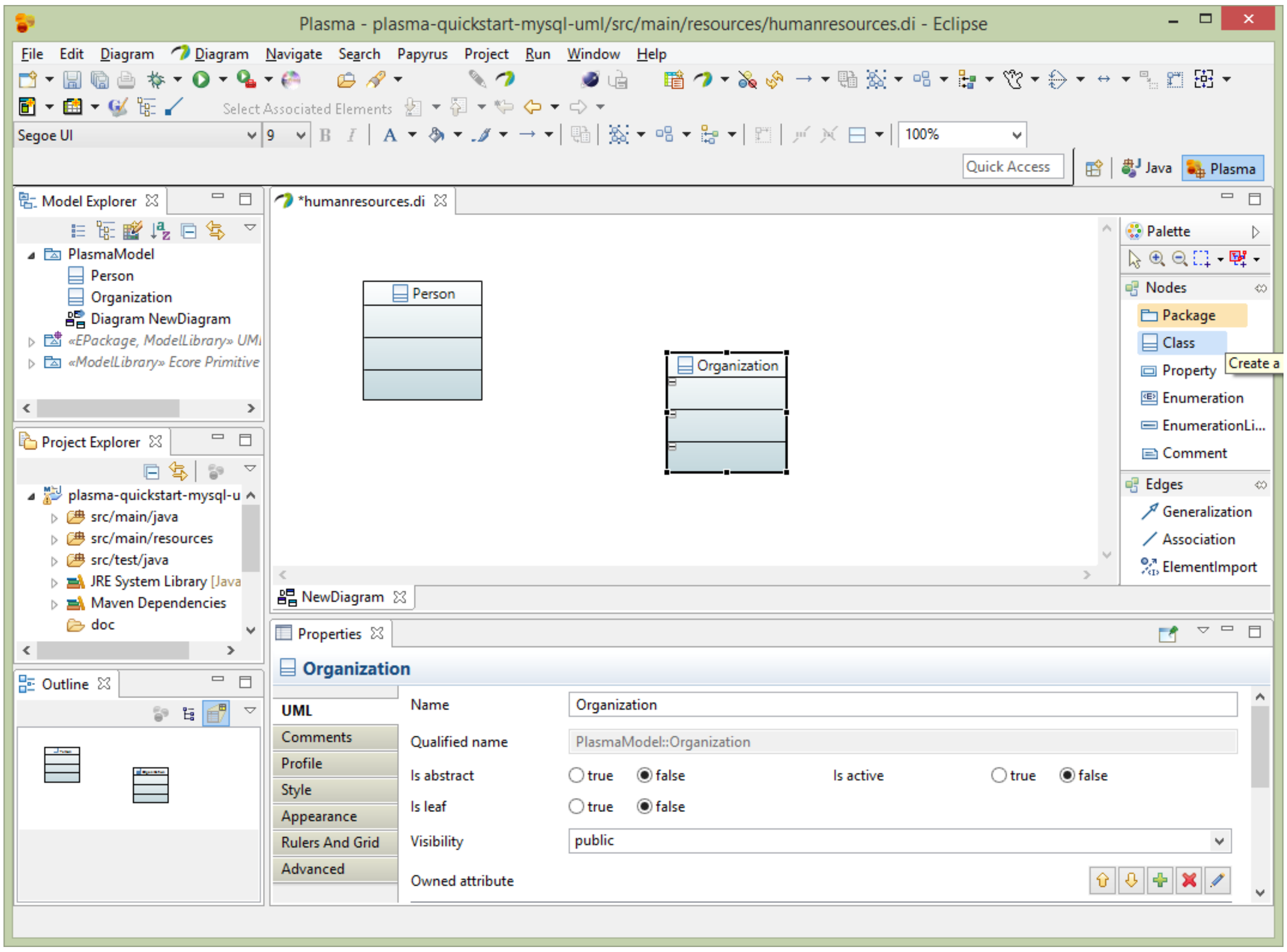


| | |
|---|---|
| Name | Organization |
| Qualified name | PlasmaModel::Organization |
| Is abstract | true [✓ false] |
| Is active | true [✓ false] |
| Is leaf | true [✓ false] |
| Visibility | public |
| Owned attribute | |

## 5 Editing a Plasma UML Model

First open the Plasma Perspective. The new model can be edited and annotated as any Papyrus model. See https://www.eclipse.org/papyrus/ for detailed documentation on Eclipse Papyrus. Plasma Data Studio however has a custom pallete which restricts the available UML elements to those applicable to Plasma structural models. Double click or drag 2 class elements to the diagram and name one Person and one Organization.

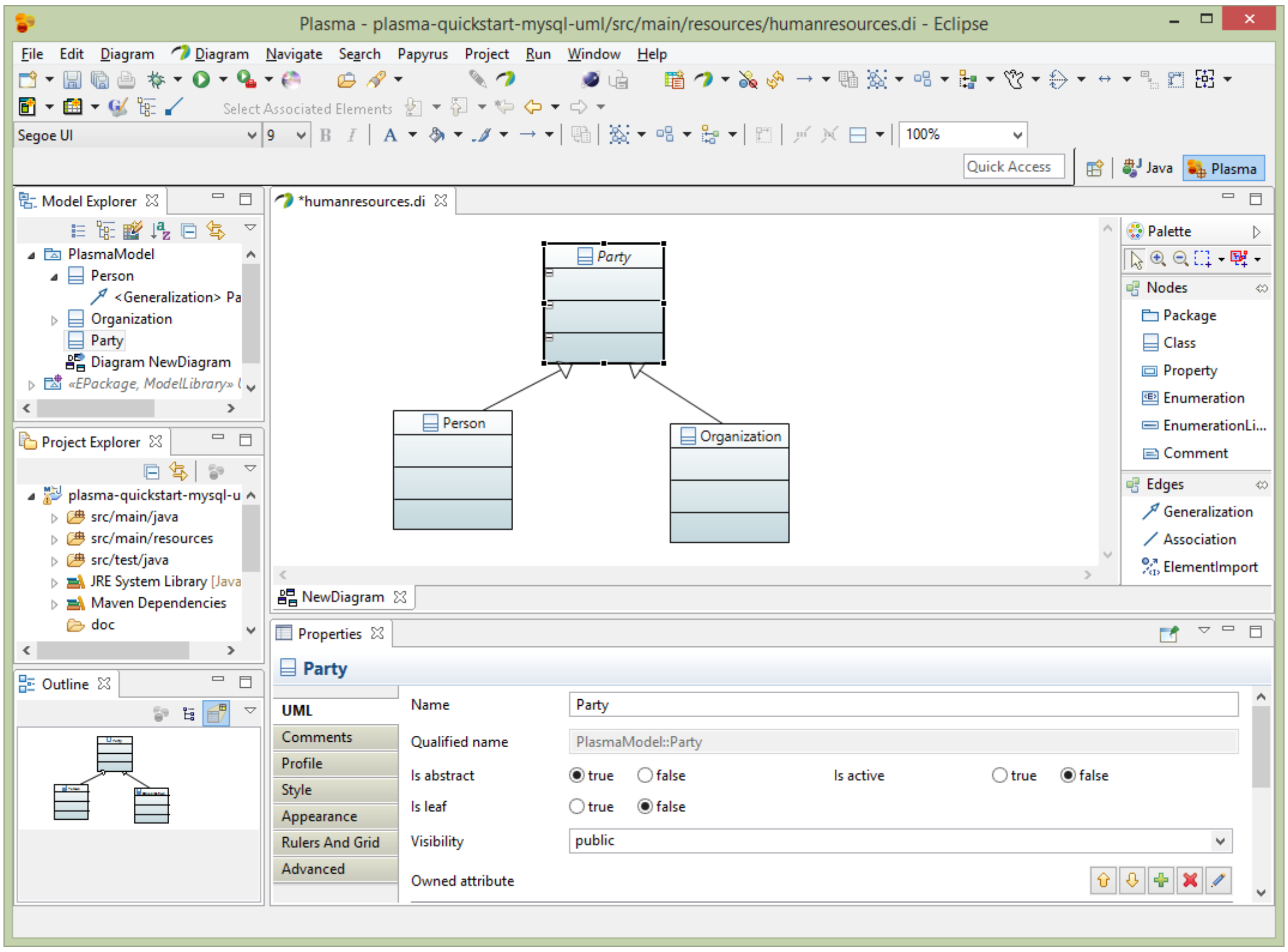




Organization

| | |
|---|---|
| Name | Organization |
| Qualified name | PlasmaModel::Organization |
| Is abstract | true [✓ false] |
| Is active | true [✓ false] |
| Is leaf | true [✓ false] |
| Visibility | public |
| Owned attribute | |

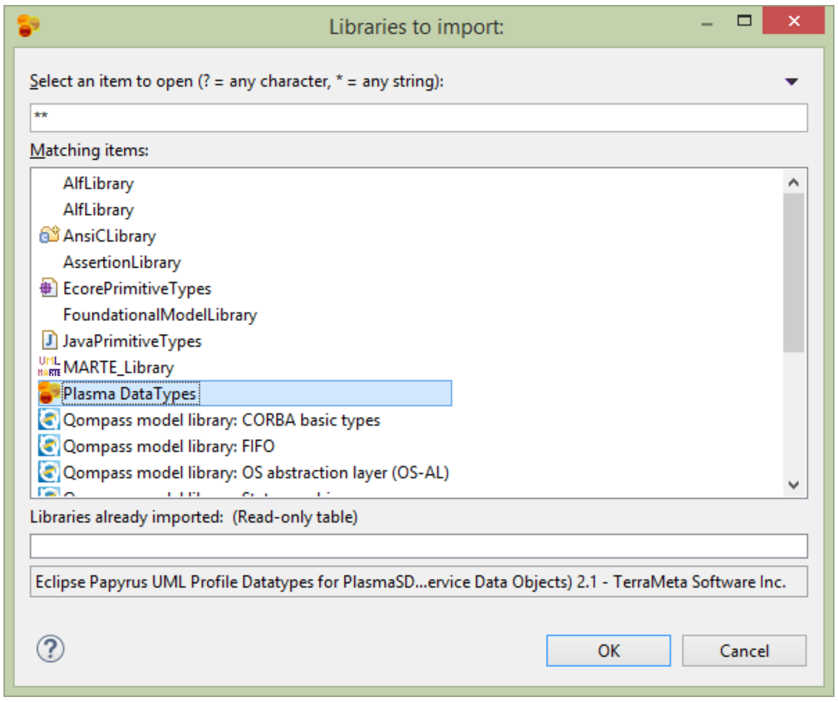
Next add a class called Party and make it anstract, then use generalization edges to make the Person and Organization classes inherit from Party.



Party

| | | | |
|---|---|---|---|
| Name | Party | | |
| Qualified name | PlasmaModel::Party | | |
| Is abstract | [✓ true] false | Is active | true [✓ false] |
| Is leaf | true [✓ false] | | |
| Visibility | public | | |
| Owned attribute | | | |

Next add properties to the Person entity. Before adding properties we need to import datatypes to use for the property datatypes. Select the root element of the project in the Model Explorer, then select Import->Import Registered Package. Then select Plasma Data Types and load the package into your model.


Libraries to import:

Select an item to open (? = any character, * = any string):

**

Matching items:

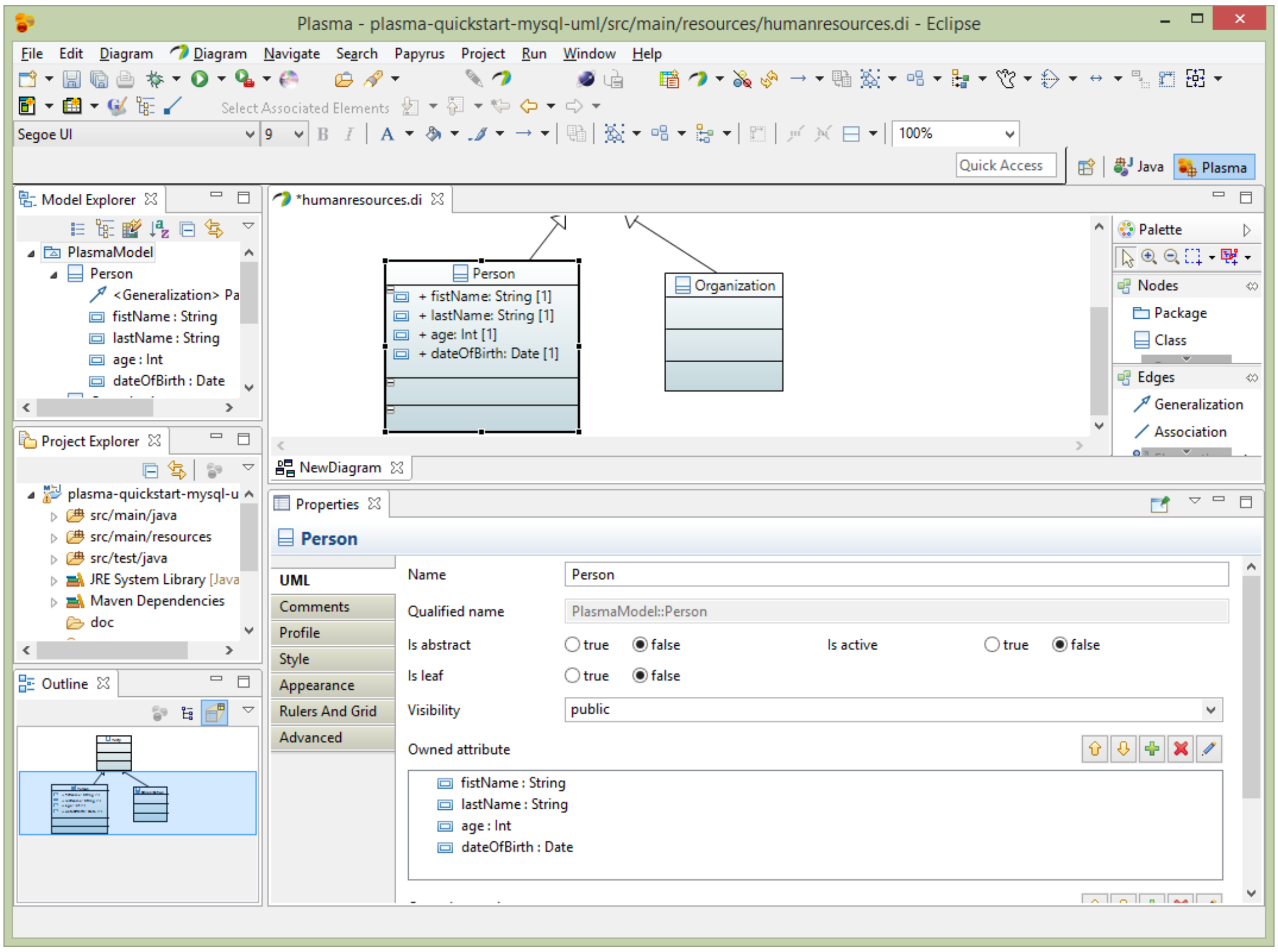
- AlfLibrary
- AlfLibrary
- AnsiCLibrary
- AssertionLibrary
- EcorePrimitiveTypes
- FoundationalModelLibrary
- JavaPrimitiveTypes
- MARTE_Library
- Plasma DataTypes
- Qompass model library: CORBA basic types
- Qompass model library: FIFO
- Qompass model library: OS abstraction layer (OS-AL)

Libraries already imported: (Read-only table)

Eclipse Papyrus UML Profile Datatypes for PlasmaSD...ervice Data Objects) 2.1 - TerraMeta Software Inc.

OK Cancel

Next using the plus (+) sign in the properties editor, add 4 properties firstName, lastName, age and dateOfBirth with datatypes (String, String, Int, Date) respectively. Note that the Papyres Filters must be used to make the properties appear in the entity. Right click on the entity and select Filters->Show/Hide Contents.


Plasma - plasma-quickstart-mysql-uml/src/main/resources/humanresources.di - Eclipse

Model Explorer:

- PlasmaModel
  - Person
    - <Generalization> Pa
    - fistName : String
    - lastName : String
    - age : Int
    - dateOfBirth : Date

*humanresources.di

Person
- + fistName: String [1]
- + lastName: String [1]
- + age: Int [1]
- + dateOfBirth: Date [1]

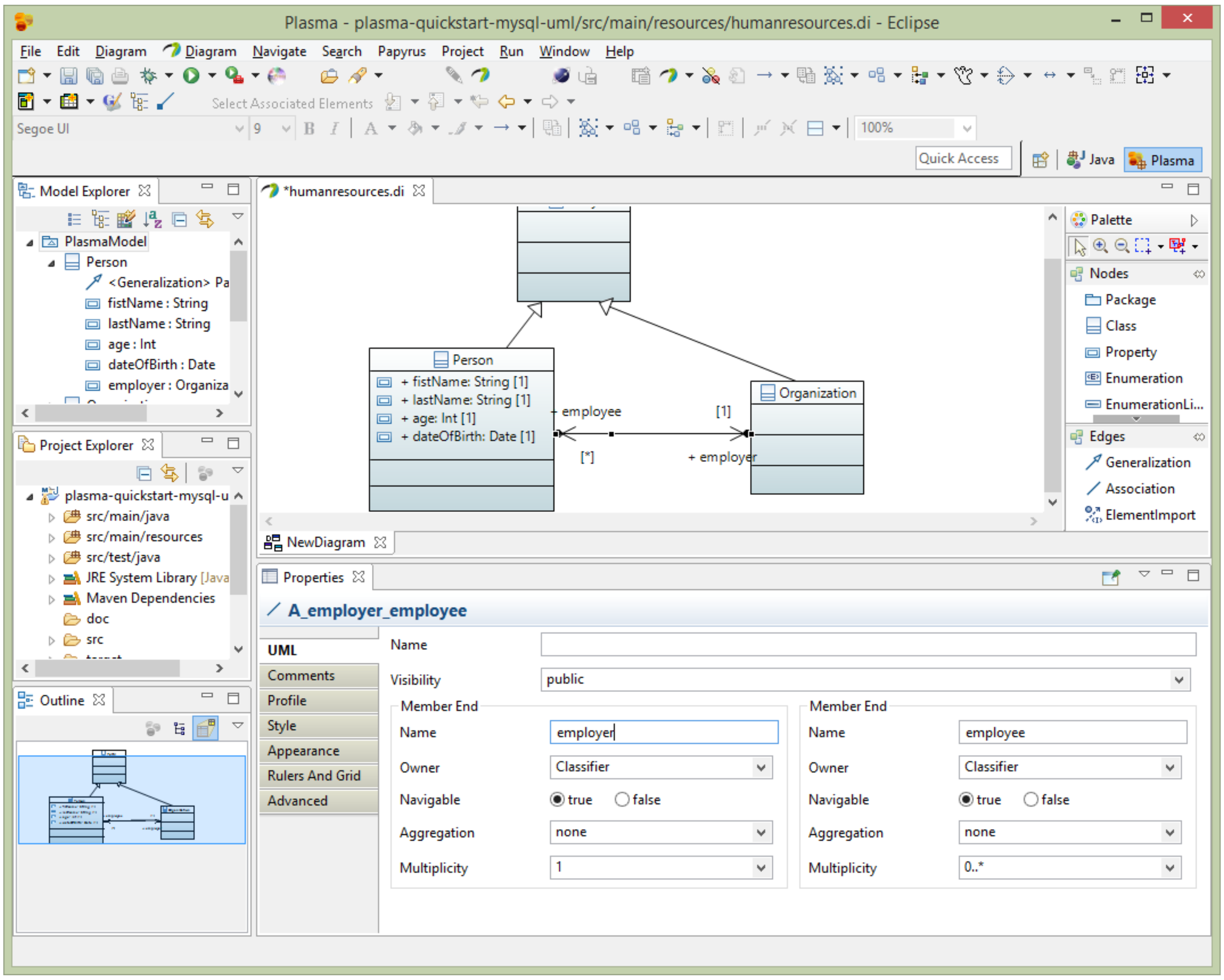
Organization

Properties — Person

| | | | | |
|---|---|---|---|---|
| Name | Person | | | |
| Qualified name | PlasmaModel::Person | | | |
| Is abstract | ○ true [✓ false] | | Is active | ○ true [✓ false] |
| Is leaf | ○ true [✓ false] | | | |
| Visibility | public | | | |

Owned attribute
- fistName : String
- lastName : String
- age : Int
- dateOfBirth : Date

Next we will link the Person and Organization entities with an association. Select Association from the edges palette and link the Person and Organization entities. Make both ends of the association navigable and owned by the respective Classifier and make the Multiplicity on the Organization side zero-to-many as below. And instead of leaving the default names for the association ends (person, organization) make these more specific to a Human Resources model such as employer and employee, as below.

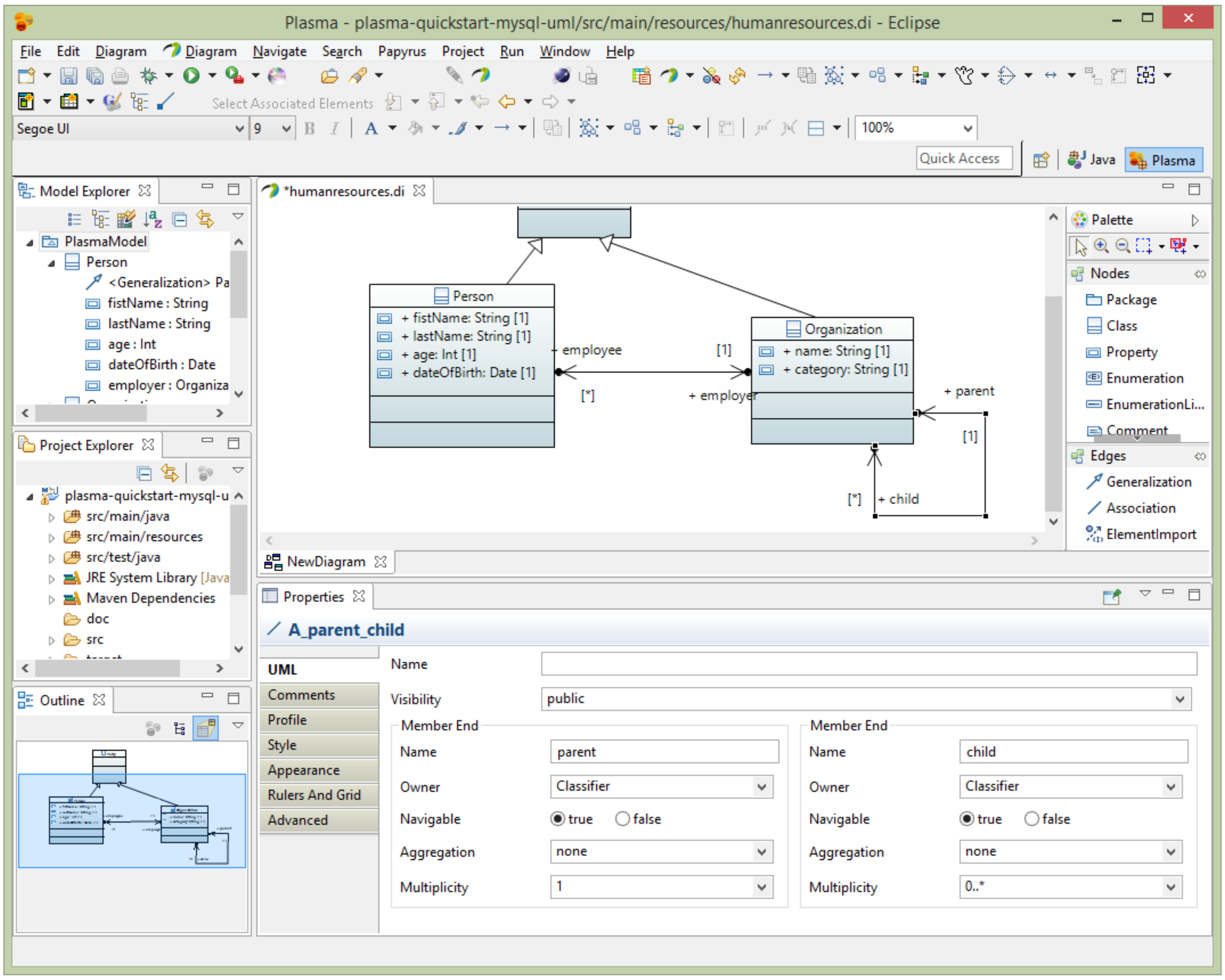


A_employer_employee

| | |
|---|---|
| Name | |
| Visibility | public |

| Member End | | Member End | |
|---|---|---|---|
| Name | employer | Name | employee |
| Owner | Classifier | Owner | Classifier |
| Navigable | [✓ true] false | Navigable | [✓ true] false |
| Aggregation | none | Aggregation | none |
| Multiplicity | 1 | Multiplicity | 0..* |

Next add 2 properties to the Organization entity name and category, both with String datatypes from the Plasma Data Types package. We will enhance the category property with an enumeration restriction later. Then add a recursive association to the Organization entity as below, with both sides of the association owned by the classifier. Note that the Papyres Filters must be used to make the properties appear in the entity. Right click on the entity and select Filters->Show/Hide Contents.



Properties

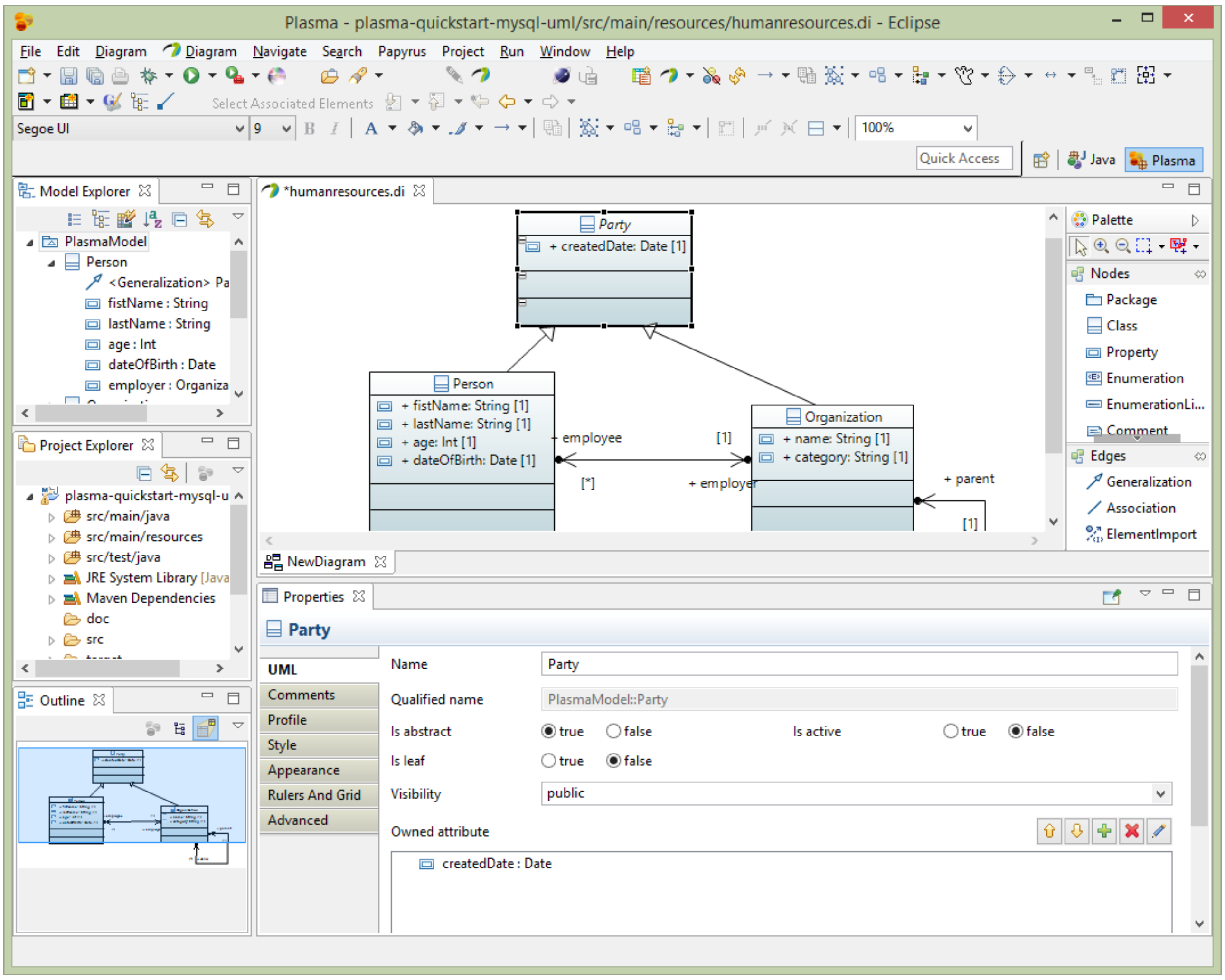
A_parent_child

UML | Comments | Profile | Style | Appearance | Rulers And Grid | Advanced

| | |
|---|---|
| Name | |
| Visibility | public |

| Member End | | Member End | |
|---|---|---|---|
| Name | parent | Name | child |
| Owner | Classifier | Owner | Classifier |
| Navigable | [✓ true] false | Navigable | [✓ true] false |
| Aggregation | none | Aggregation | none |
| Multiplicity | 1 | Multiplicity | 0..* |

Nexte add a property to the shared entity we created previously called Party. Just add 1 property createdDate of data type Date. This property will be inherited by both the Person and Organization entities.



Properties

Party

UML | Comments | Profile | Style | Appearance | Rulers And Grid | Advanced

Name: Party

Qualified name: PlasmaModel::Party

Is abstract: [✓ true] false

Is active: true [✓ false]

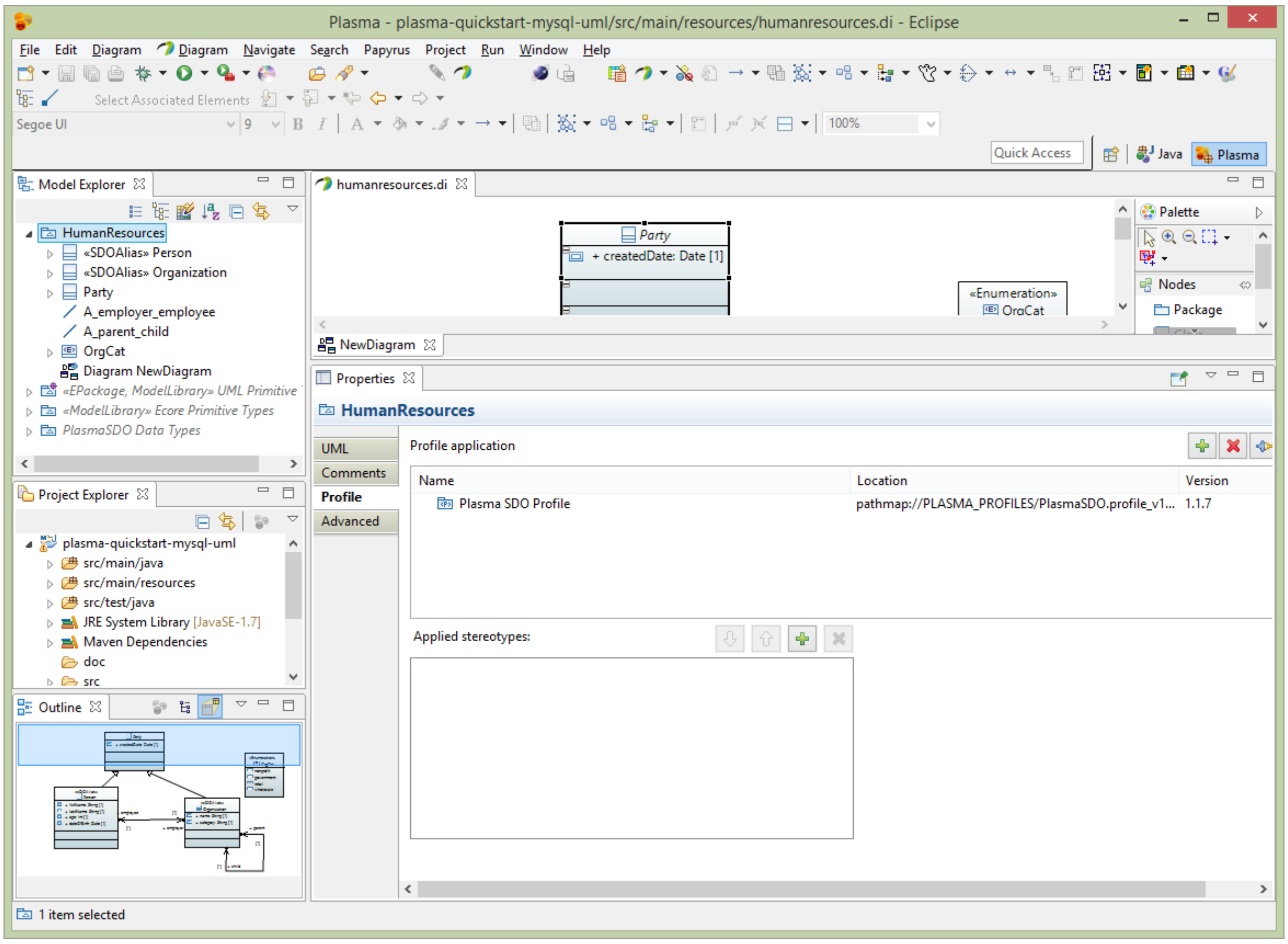
Is leaf: true [✓ false]

Visibility: public

Owned attribute: createdDate : Date

# 6 Enhancing a Plasma UML Model

After creating a basic model we need to add enhancements using standard UML Profile mechanisms supported by Eclipse Papyrus, which add “facets” to various elements. These facets describe restrictions such as String property length restrictions enumeration restrictions and several others which provide a complete structural model description. The final model has enough information to allow us to generate code and persist the entities we created in one or more physical data stores. (*Note that for big-data stores such as HBase the size of the physical table is greatly influenced by the size of column names, and the logical/physical name isolation provided by Plasma is critical for controlling the size of the physical store while maintaining a readable data model and generated code*) First select the root package node on the Model Explorer and rename the package to HumanResources. Then select the Profile editor section and add the SDOAlias and SDONamespace stereotypes. Assign the SDOAlias physicalName = ‘’HR’ for the model package and the SDONamespace uri = ‘`http://plasma-quickstart-pojo/humanresources`’. The URI is critical and is used to link the model to a specific data store. A large model can of course have numerous packages and each package can be associated the the same or several data store providers.


Plasma - plasma-quickstart-mysql-uml/src/main/resources/humanresources.di - Eclipse
File Edit Diagram Diagram Navigate Search Papyrus Project Run Window Help
Select Associated Elements
Segoe UI 9
100%
Quick Access
Java
Plasma
Model Explorer
humanresources.di
Palette
Nodes
Package
NewDiagram
Properties
Outline
Project Explorer

Model Explorer:
- HumanResources
  - «SDOAlias» Person
  - «SDOAlias» Organization
  - Party
  - A_employer_employee
  - A_parent_child
  - OrgCat
  - Diagram NewDiagram
- «EPackage, ModelLibrary» UML Primitive
- «ModelLibrary» Ecore Primitive Types
- PlasmaSDO Data Types

Diagram: Party — + createdDate: Date [1]; «Enumeration» OrgCat

Project Explorer:
- plasma-quickstart-mysql-uml
  - src/main/java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-1.7]
  - Maven Dependencies
  - doc
  - src

Properties — HumanResources

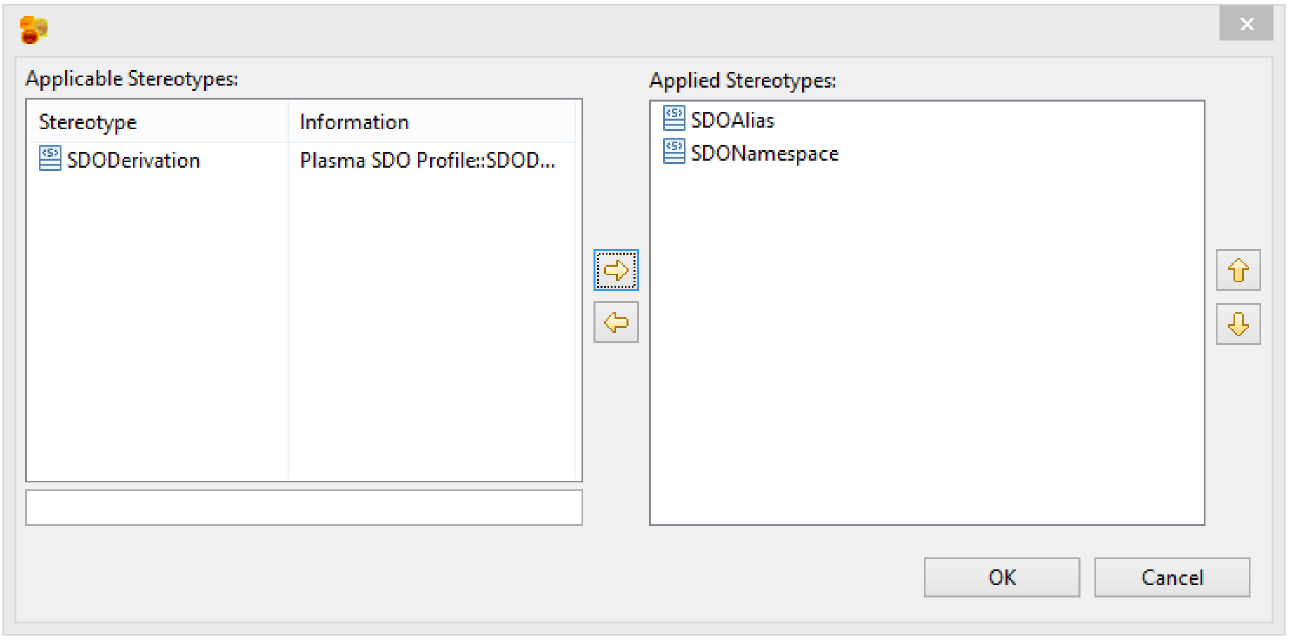
Tabs: UML, Comments, Profile, Advanced

Profile application

| Name | Location | Version |
|---|---|---|
| Plasma SDO Profile | pathmap://PLASMA_PROFILES/PlasmaSDO.profile_v1... | 1.1.7 |

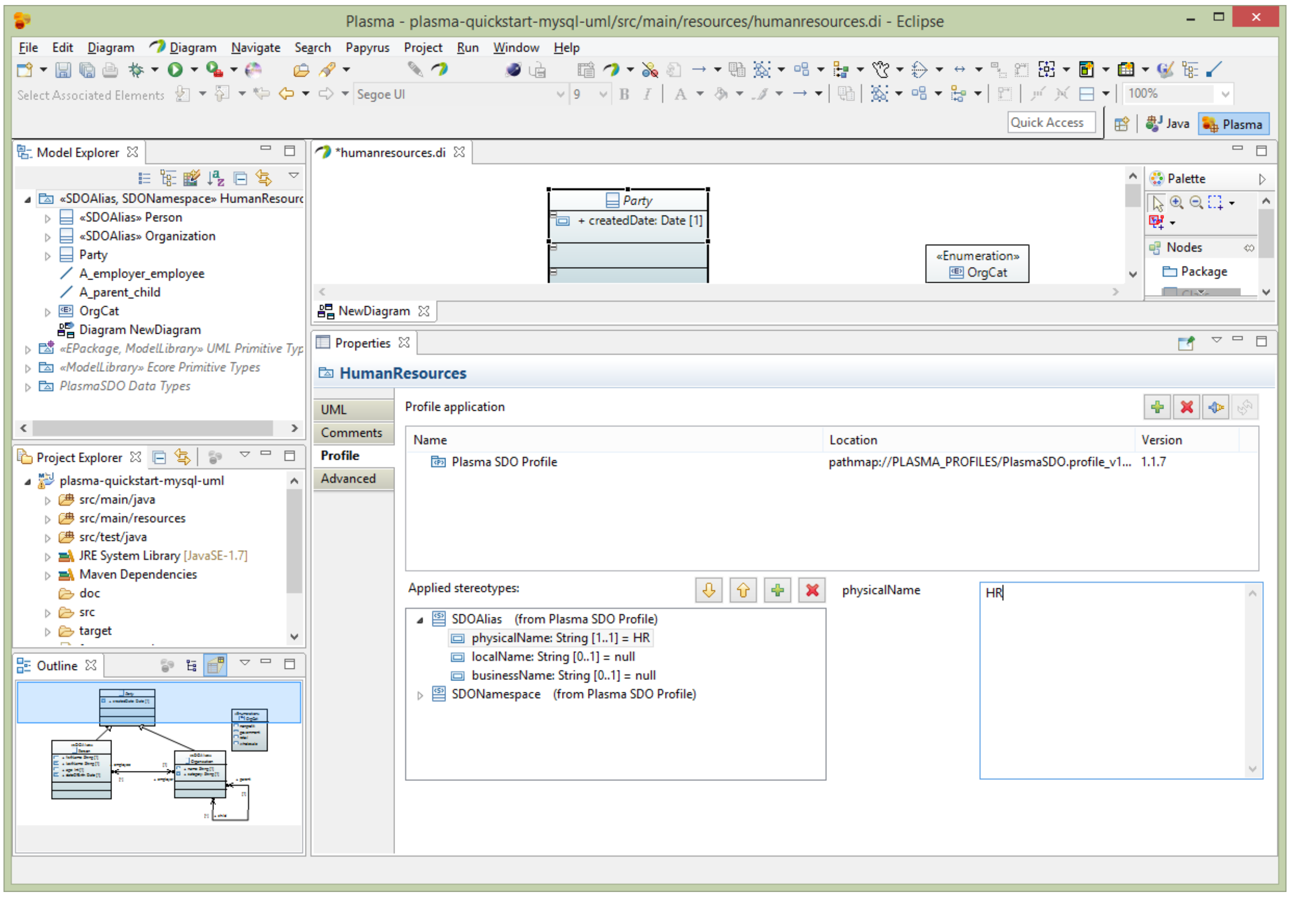
Applied stereotypes:

1 item selected


Applicable Stereotypes:

| Stereotype | Information |
|---|---|
| SDODerivation | Plasma SDO Profile::SDOD... |

Applied Stereotypes:
- SDOAlias
- SDONamespace



«SDOAlias, SDONamespace» HumanResourc
- «SDOAlias» Person
- «SDOAlias» Organization
- Party
- A_employer_employee
- A_parent_child
- OrgCat
- Diagram NewDiagram

«EPackage, ModelLibrary» UML Primitive Typ
«ModelLibrary» Ecore Primitive Types
PlasmaSDO Data Types

plasma-quickstart-mysql-uml
- src/main/java
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-1.7]
- Maven Dependencies
- doc
- src
- target

**HumanResources**

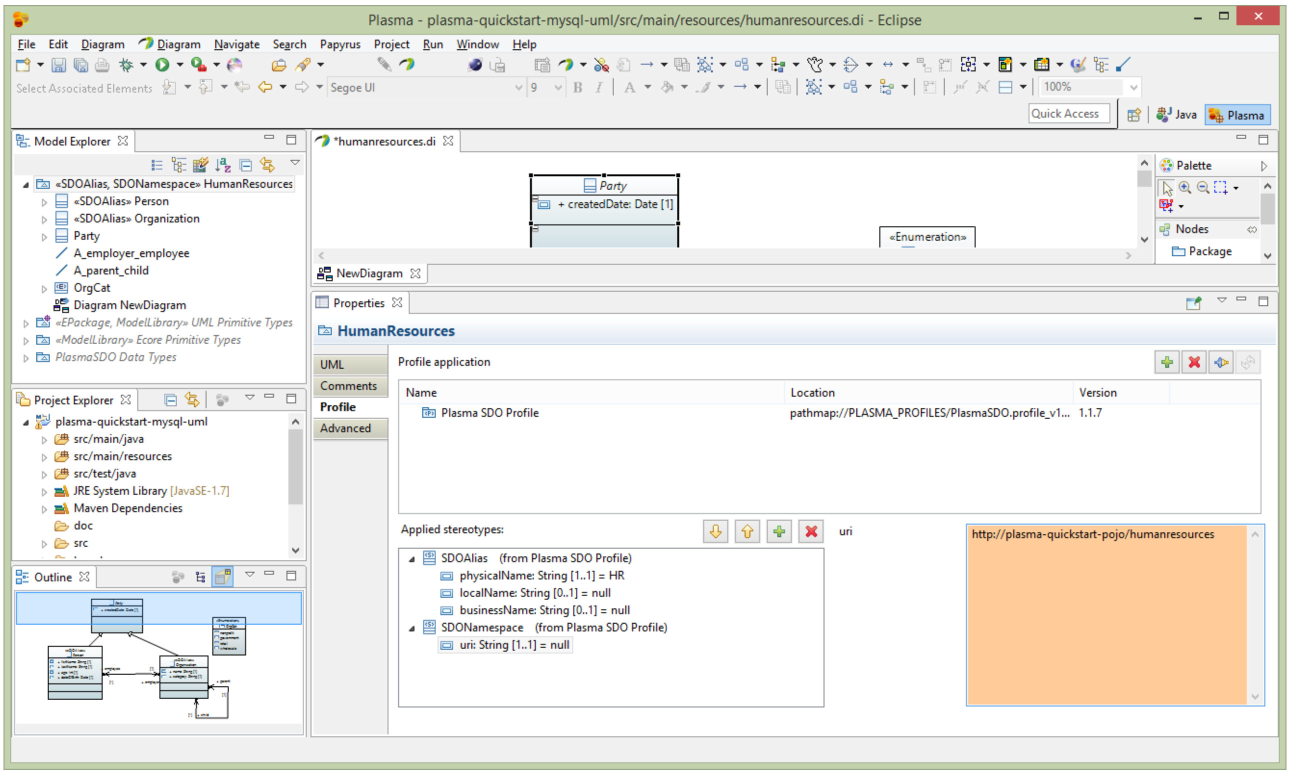
Profile application

| Name | Location | Version |
|---|---|---|
| Plasma SDO Profile | pathmap://PLASMA_PROFILES/PlasmaSDO.profile_v1... | 1.1.7 |

Applied stereotypes:

- SDOAlias (from Plasma SDO Profile)
  - physicalName: String [1..1] = HR
  - localName: String [0..1] = null
  - businessName: String [0..1] = null
- SDONamespace (from Plasma SDO Profile)

physicalName: HR



«SDOAlias, SDONamespace» HumanResources
- «SDOAlias» Person
- «SDOAlias» Organization
- Party
- A_employer_employee
- A_parent_child
- OrgCat
- Diagram NewDiagram

«EPackage, ModelLibrary» UML Primitive Types
«ModelLibrary» Ecore Primitive Types
PlasmaSDO Data Types

plasma-quickstart-mysql-uml
- src/main/java
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-1.7]
- Maven Dependencies
- doc
- src

**HumanResources**

Profile application

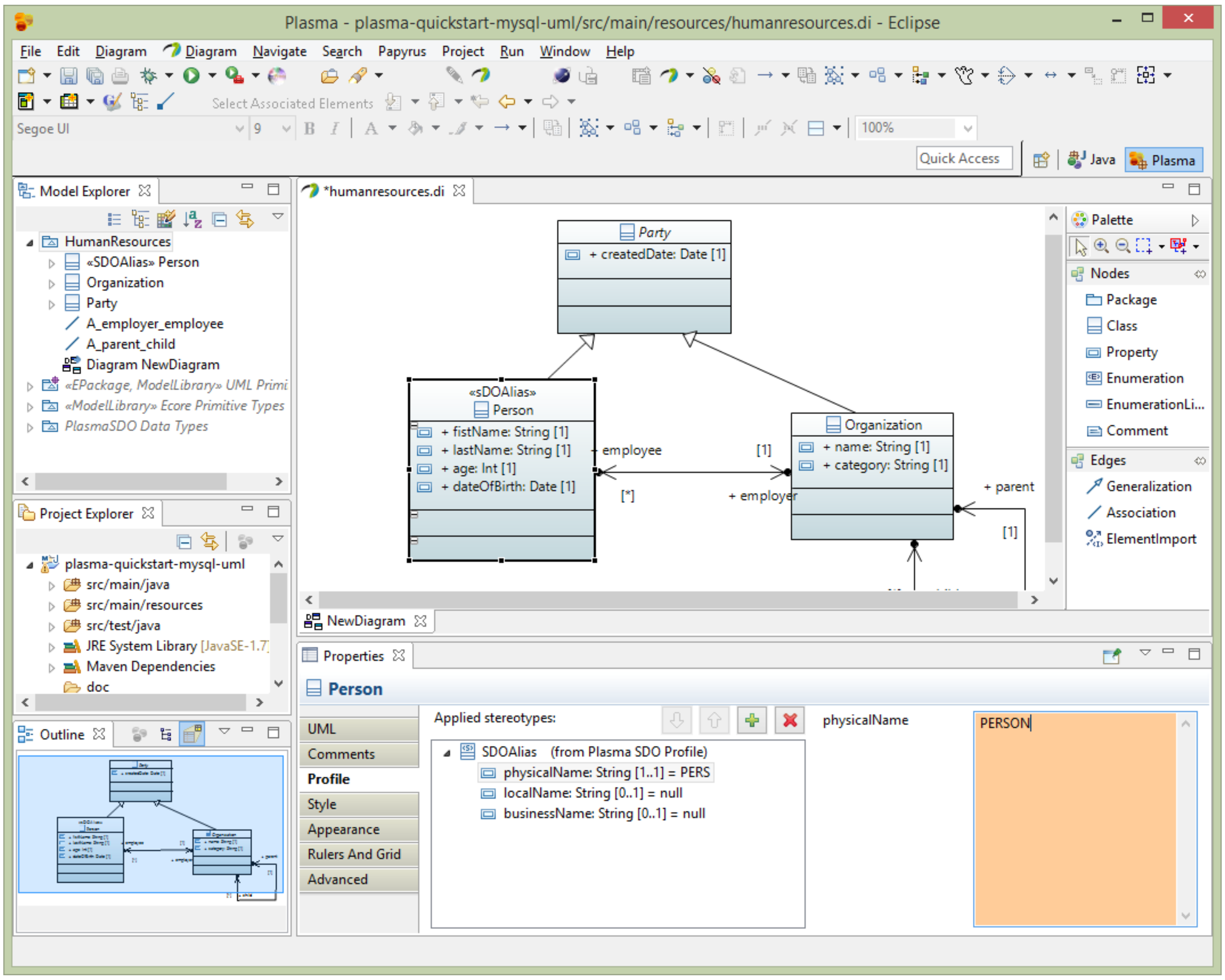
| Name | Location | Version |
|---|---|---|
| Plasma SDO Profile | pathmap://PLASMA_PROFILES/PlasmaSDO.profile_v1... | 1.1.7 |

Applied stereotypes:

- SDOAlias (from Plasma SDO Profile)
  - physicalName: String [1..1] = HR
  - localName: String [0..1] = null
  - businessName: String [0..1] = null
- SDONamespace (from Plasma SDO Profile)
  - uri: String [1..1] = null

uri: http://plasma-quickstart-pojo/humanresources

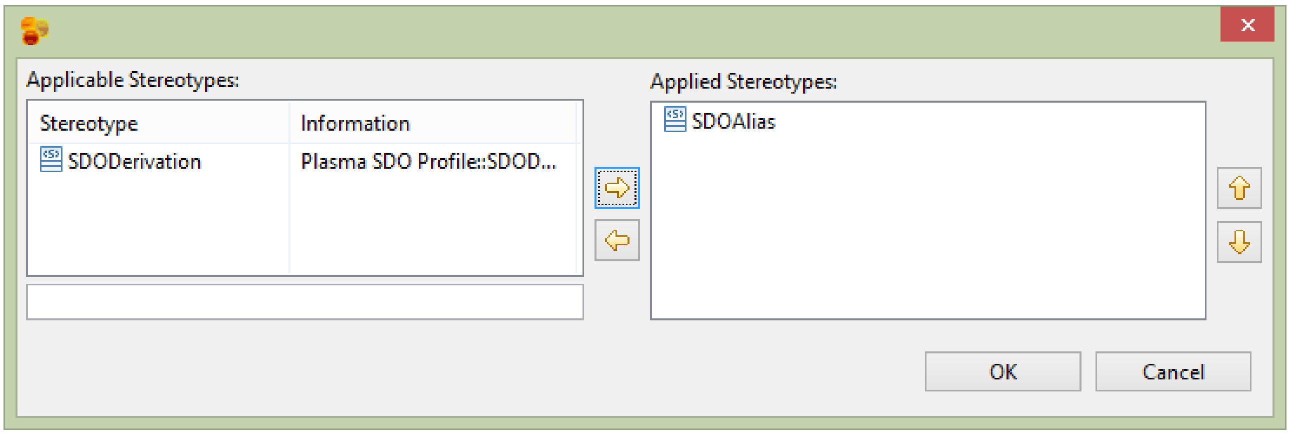
Next select the Person entity and edit the Profile section of the property editor as below. Add the SDOAlias stereotype and assign the physicalName as ‘PERSON’.



Applied stereotypes:

SDOAlias (from Plasma SDO Profile)
physicalName: String [1..1] = PERS
localName: String [0..1] = null
businessName: String [0..1] = null

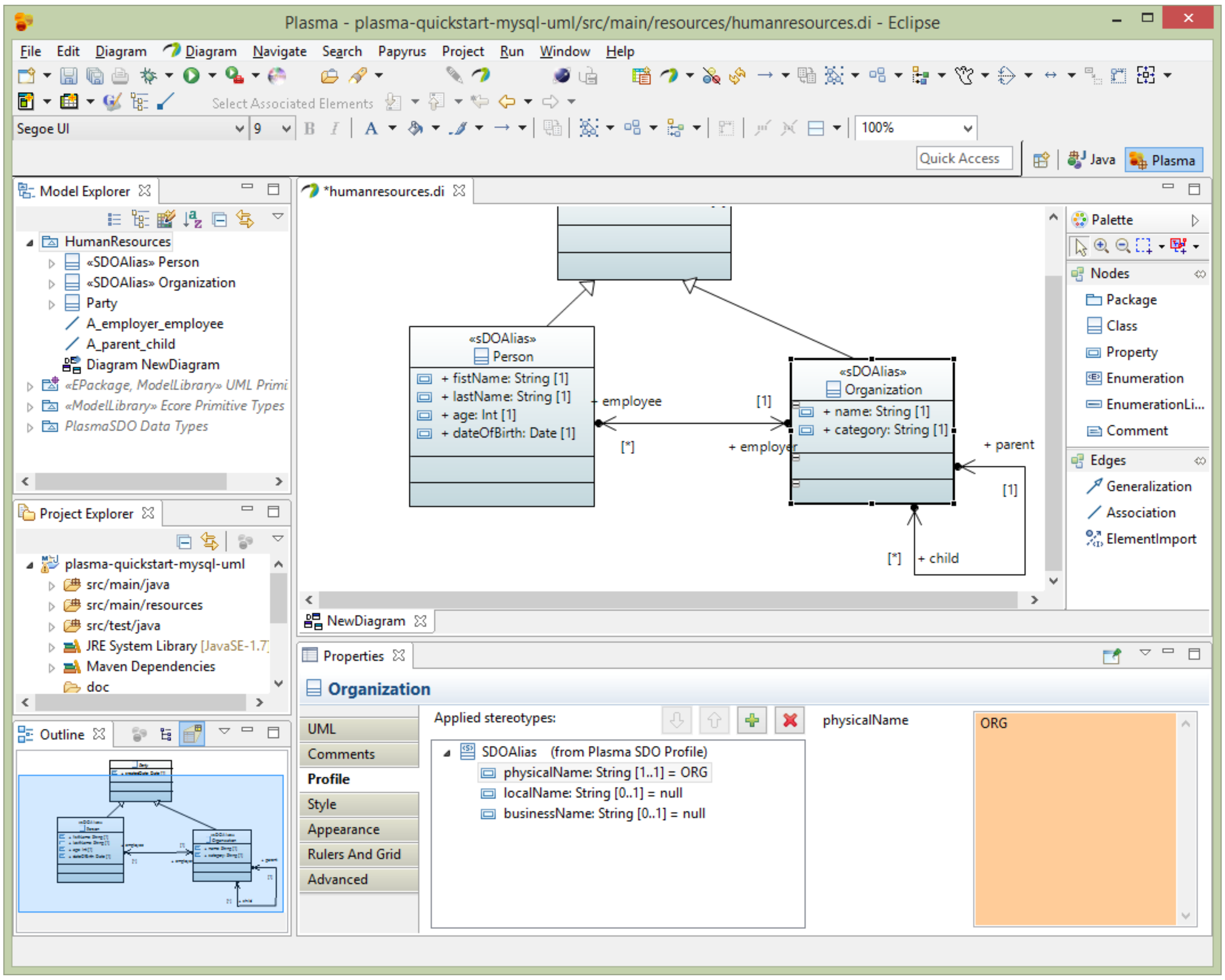
physicalName: PERSON



| Stereotype | Information |
|---|---|
| SDODerivation | Plasma SDO Profile::SDOD... |

Applied Stereotypes: SDOAlias

Next assign the Orgaization entity the physicalName of ‘ORG’.

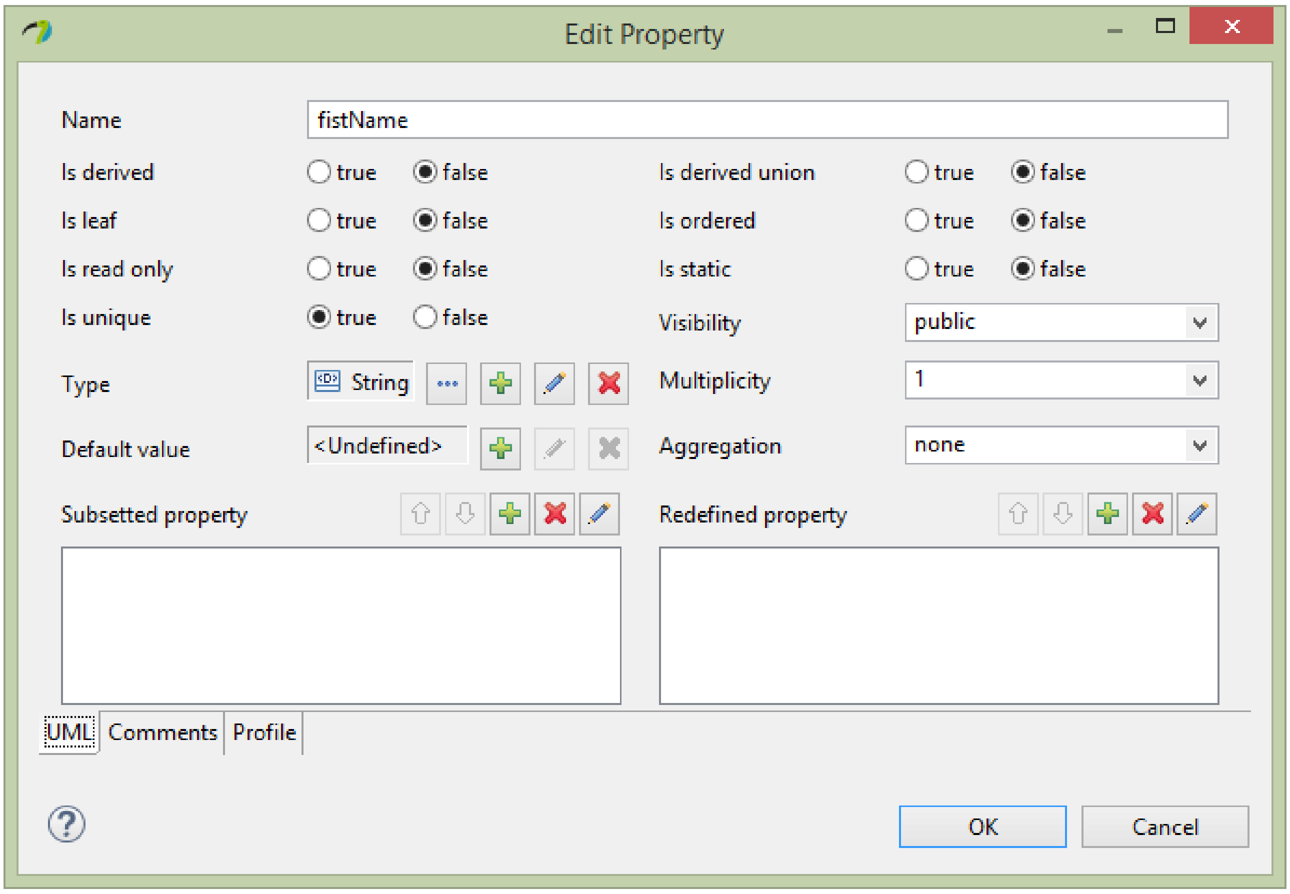


Applied stereotypes:

SDOAlias (from Plasma SDO Profile)
physicalName: String [1..1] = ORG
localName: String [0..1] = null
businessName: String [0..1] = null

physicalName ORG

Next for the Person entity firstName property, select the Profile section of the editor and assign the SDOAlias, SDOKey and SDOValueConstraint stereotypes. Then assign SDOAlias physicalName = “FN” and SDOKey type=’primary’ and SDOValueConstraint maxLength=36 as below.

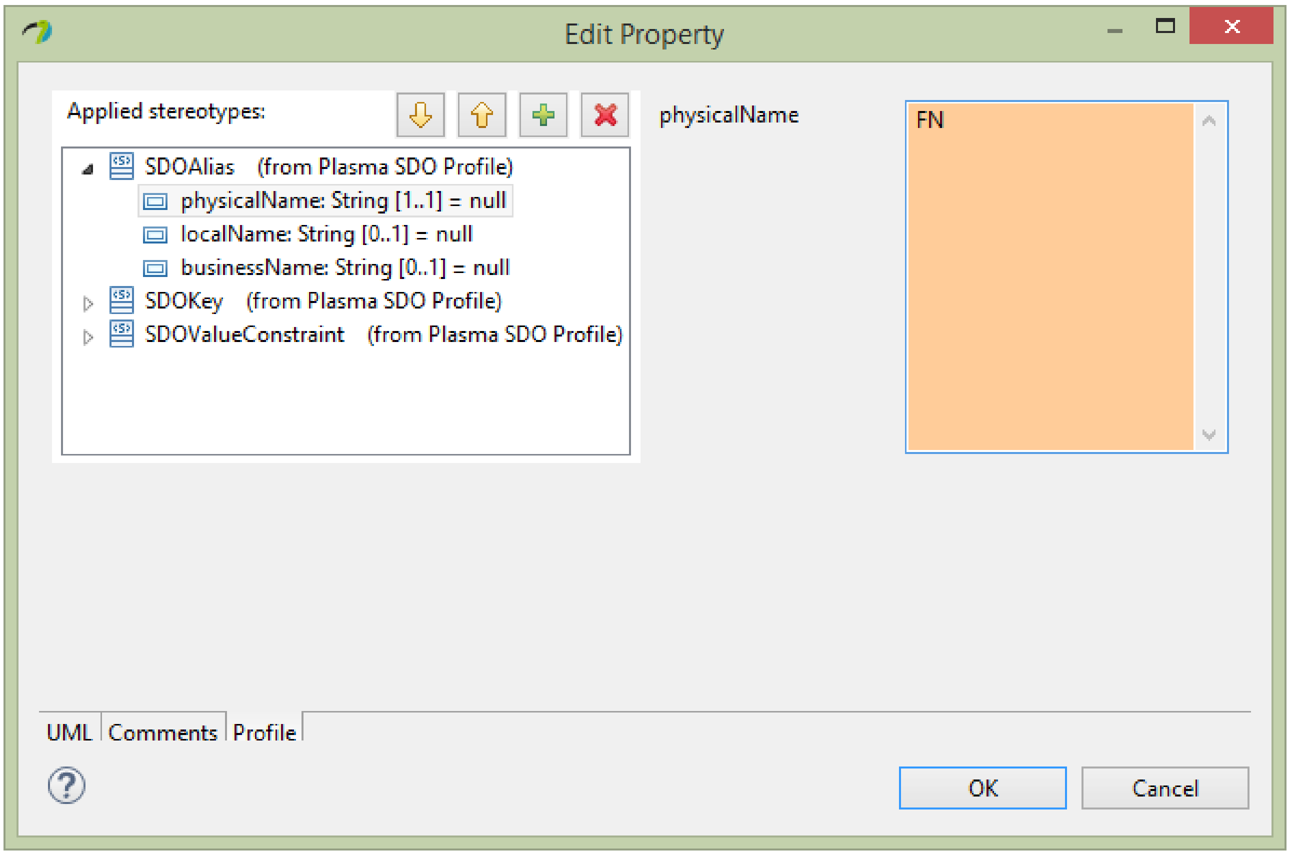


Edit Property

| | | | | | |
|---|---|---|---|---|---|
| Name | fistName | | | | |
| Is derived | true | [✓ false] | Is derived union | true | [✓ false] |
| Is leaf | true | [✓ false] | Is ordered | true | [✓ false] |
| Is read only | true | [✓ false] | Is static | true | [✓ false] |
| Is unique | [✓ true] | false | Visibility | public | |
| Type | String | | Multiplicity | 1 | |
| Default value | <Undefined> | | Aggregation | none | |

Subsetted property

Redefined property

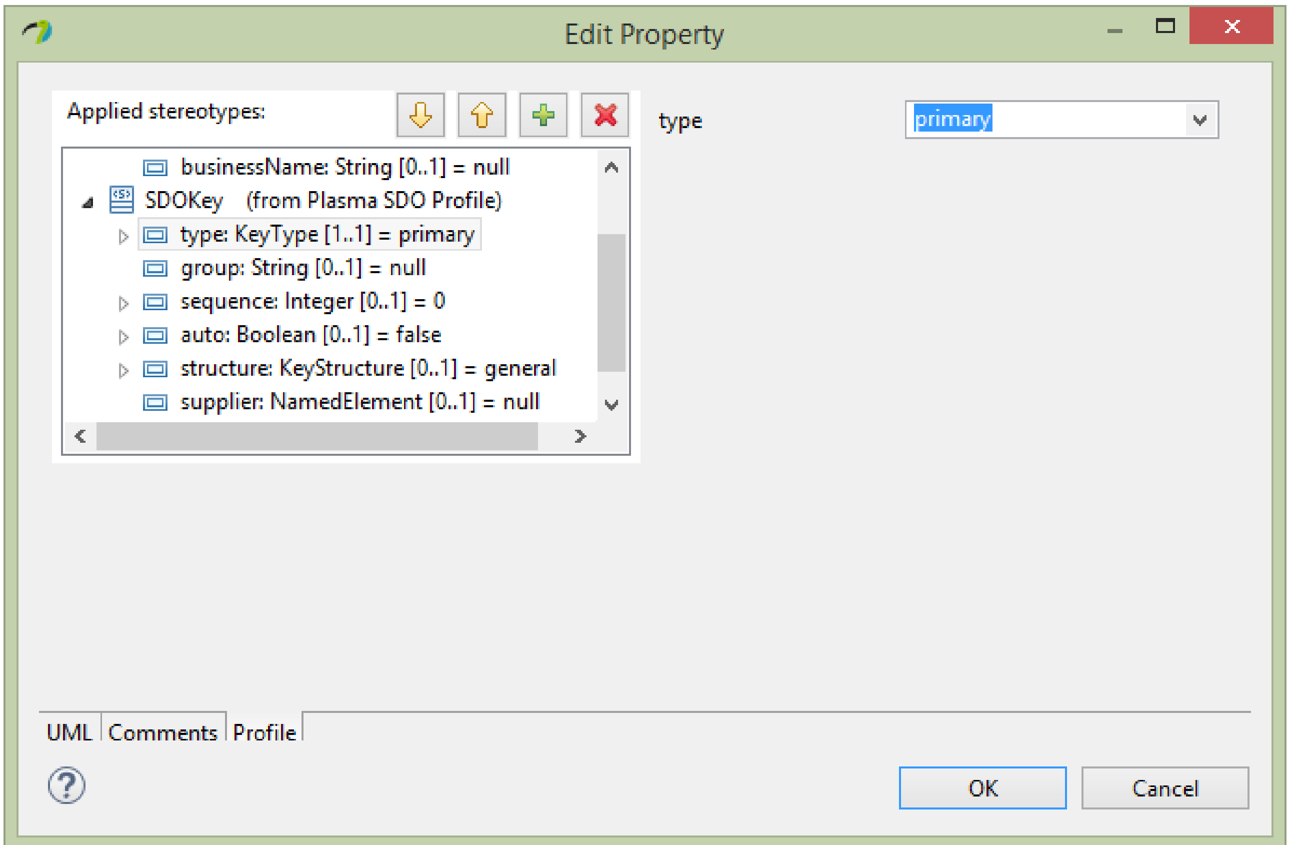


Edit Property

Applied stereotypes:

- SDOAlias (from Plasma SDO Profile)
  - physicalName: String [1..1] = null
  - localName: String [0..1] = null
  - businessName: String [0..1] = null
- SDOKey (from Plasma SDO Profile)
- SDOValueConstraint (from Plasma SDO Profile)

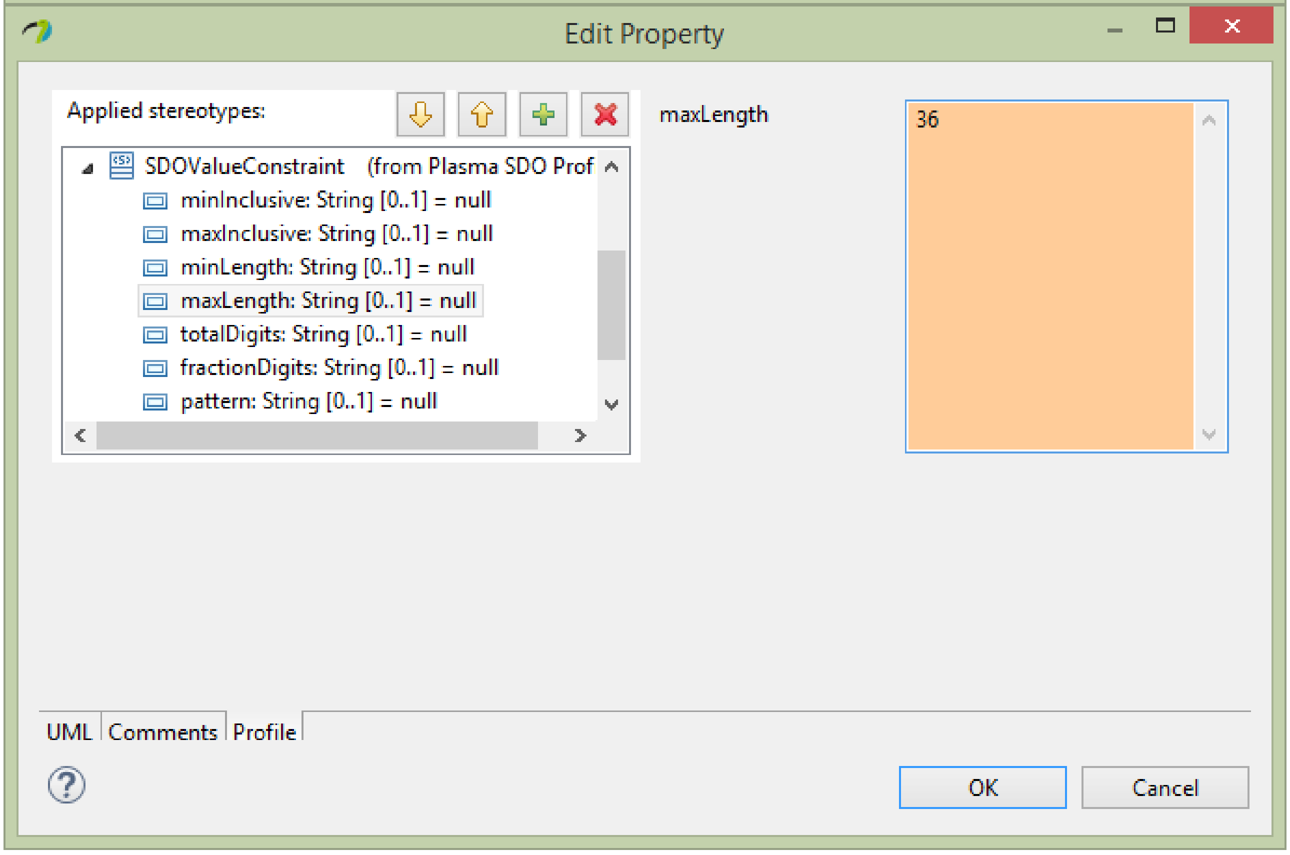
physicalName: FN


Edit Property

Applied stereotypes:

- businessName: String [0..1] = null
- SDOKey (from Plasma SDO Profile)
  - type: KeyType [1..1] = primary
  - group: String [0..1] = null
  - sequence: Integer [0..1] = 0
  - auto: Boolean [0..1] = false
  - structure: KeyStructure [0..1] = general
  - supplier: NamedElement [0..1] = null

type: primary


Edit Property

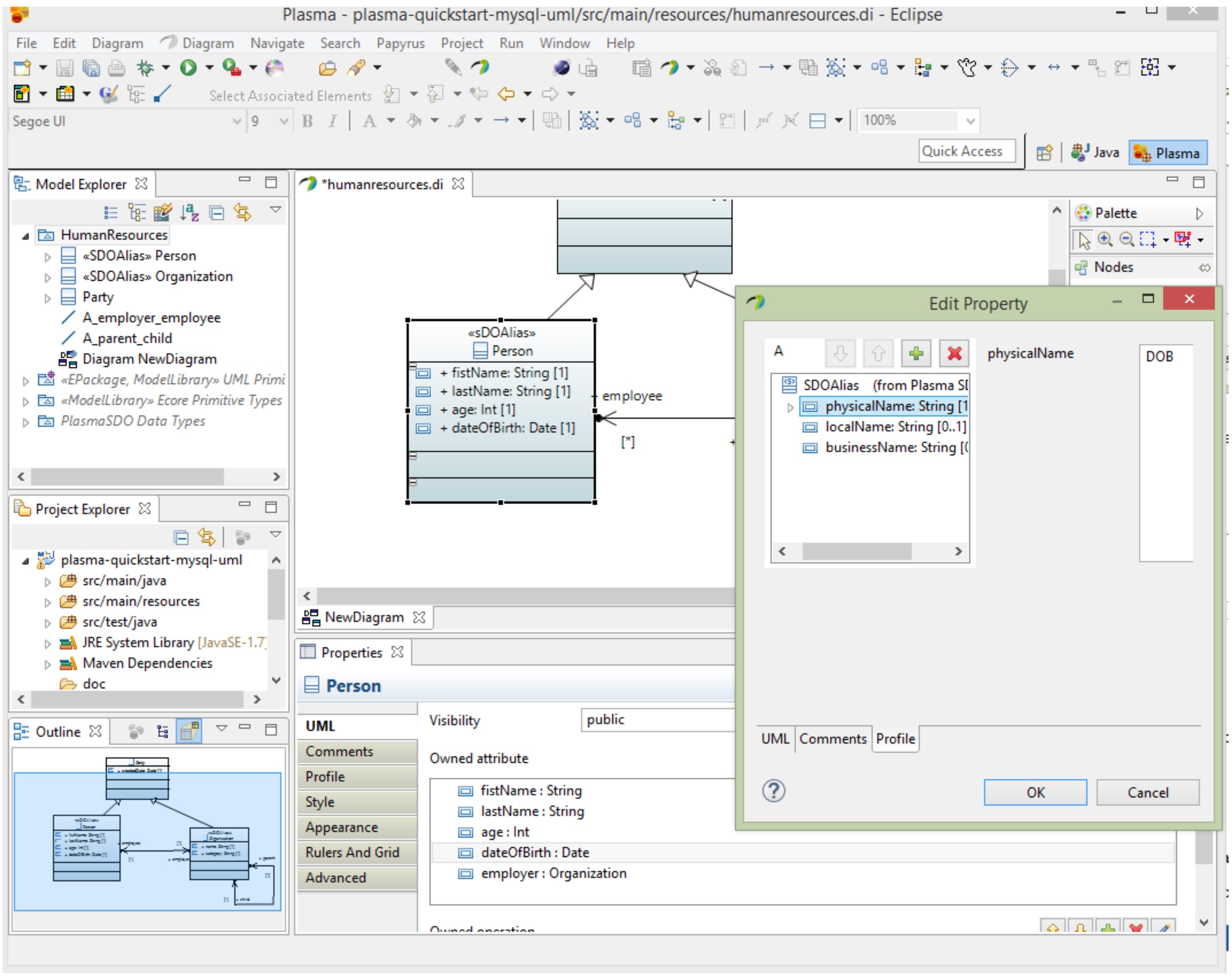
Applied stereotypes:

- SDOValueConstraint (from Plasma SDO Prof
  - minInclusive: String [0..1] = null
  - maxInclusive: String [0..1] = null
  - minLength: String [0..1] = null
  - maxLength: String [0..1] = null
  - totalDigits: String [0..1] = null
  - fractionDigits: String [0..1] = null
  - pattern: String [0..1] = null

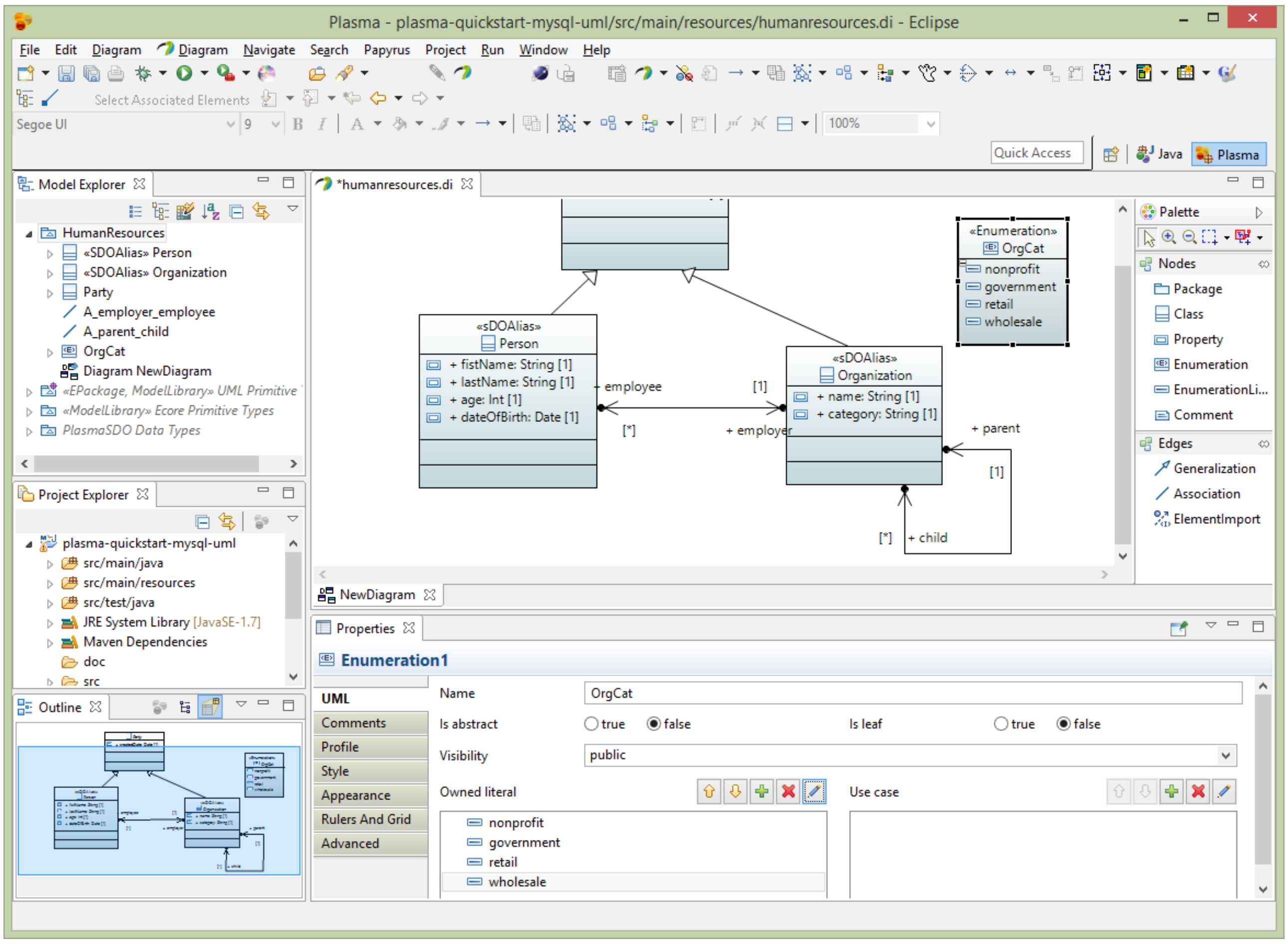
maxLength: 36

Next similarly for the Person entity lastName property, select the Profile section of the editor and assign the SDOAlias, SDOKey and SDOValueConstraint stereotypes. Then assign SDOAlias physicalName = “LN” and SDOKey type=’primary’ and SDOValueConstraint maxLength=36.

Next similarly for the Person entity dateOfBirth property, select the Profile section of the editor and assign the SDOAlias stereotype. Then assign SDOAlias physicalName = “DOB”.



Next to enhance the Organization entity, we will first create an Enumeration ‘OrgCat’ which will be used to restrict the values of the category property we added earlier. Add four owned literals to the OrgCat enumeration ‘nonprofit’, ‘government’, ‘retail’, “wholesale” and assign each literal a physicalName which is the fitst letter i.e. ‘N’, ‘G’, ‘R’ and ‘W’ respectively as below.



Name: OrgCat

Is abstract: ( ) true (●) false

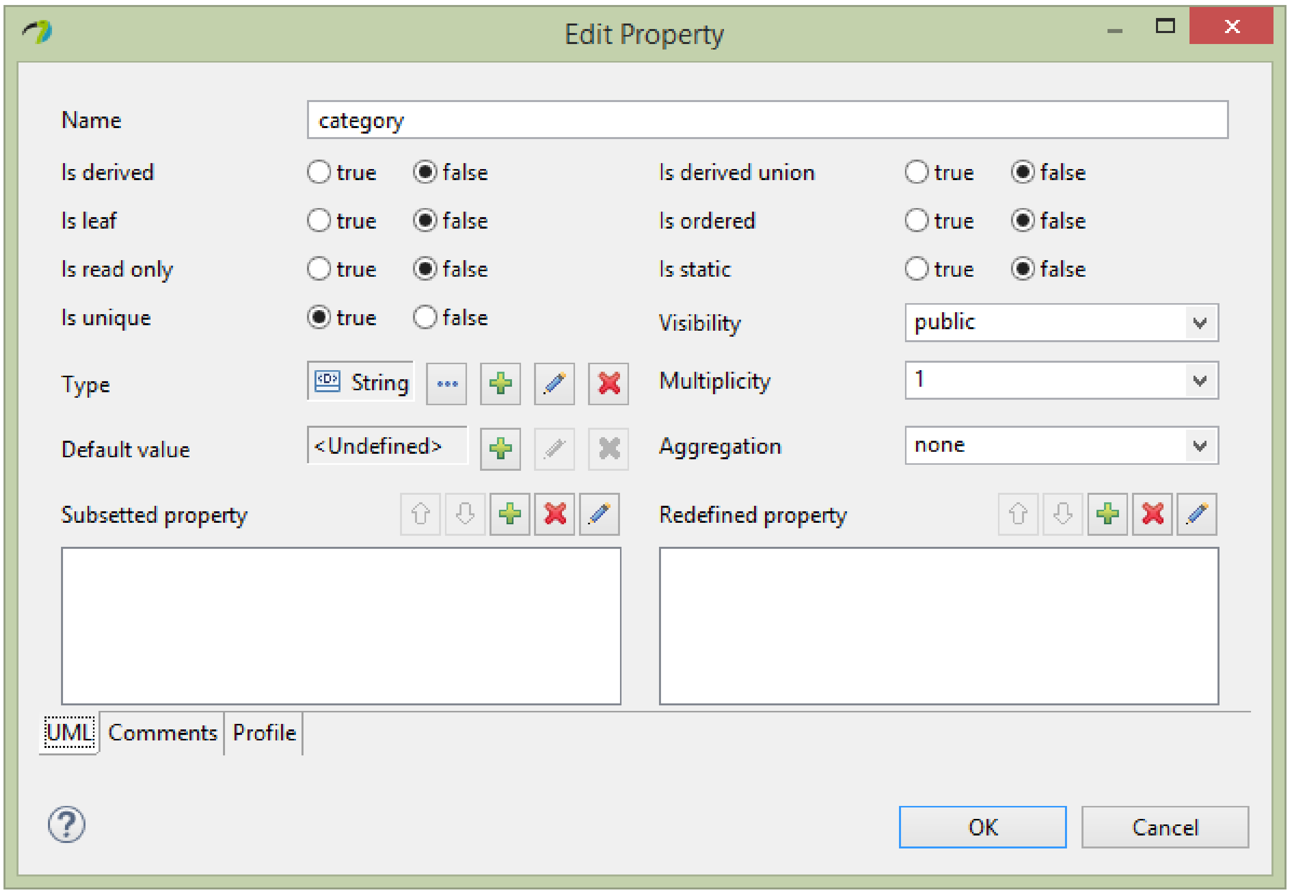
Is leaf: ( ) true (●) false

Visibility: public

Owned literal:
- nonprofit
- government
- retail
- wholesale

Use case:

Next edit the Organization category property Profile section adding the SDOAlias, SDOEnumerationConstraint and SDOValueConstraint stereotypes. Assign the SDOAlias and SDOValueConstraint stereotypes as below and assign the SDOEnumerationConstraint to the OrgCat enumeration we created previously.

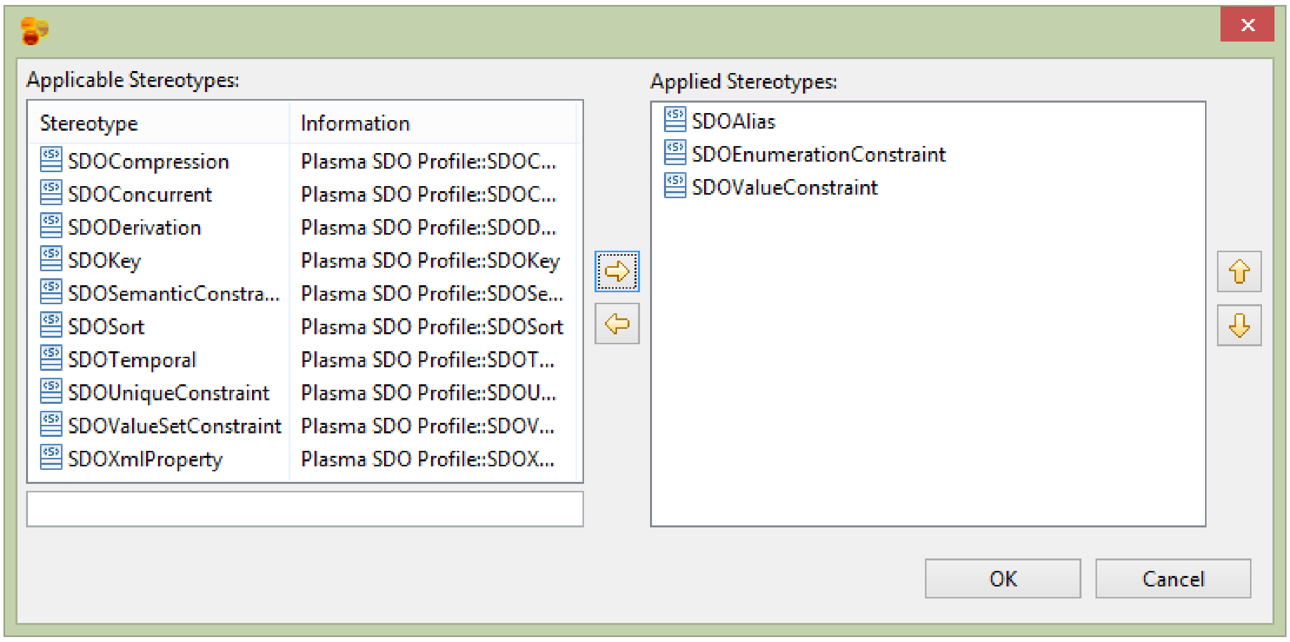


Edit Property

| | | | |
|---|---|---|---|
| Name | category | | |
| Is derived | true [✓ false] | Is derived union | true [✓ false] |
| Is leaf | true [✓ false] | Is ordered | true [✓ false] |
| Is read only | true [✓ false] | Is static | true [✓ false] |
| Is unique | [✓ true] false | Visibility | public |
| Type | String | Multiplicity | 1 |
| Default value | <Undefined> | Aggregation | none |

Subsetted property

Redefined property

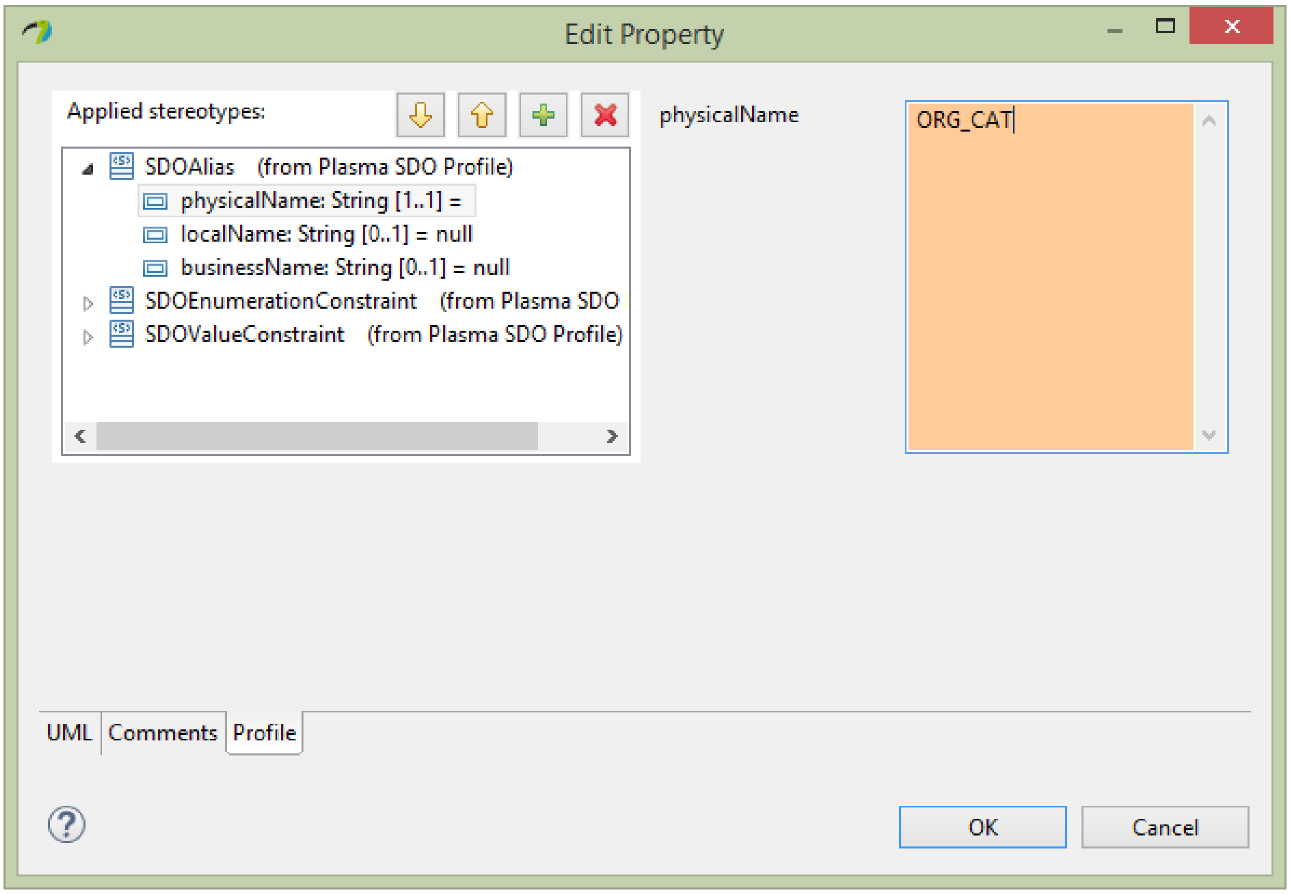


Applicable Stereotypes:

| Stereotype | Information |
|---|---|
| SDOCompression | Plasma SDO Profile::SDOC... |
| SDOConcurrent | Plasma SDO Profile::SDOC... |
| SDODerivation | Plasma SDO Profile::SDOD... |
| SDOKey | Plasma SDO Profile::SDOKey |
| SDOSemanticConstra... | Plasma SDO Profile::SDOSe... |
| SDOSort | Plasma SDO Profile::SDOSort |
| SDOTemporal | Plasma SDO Profile::SDOT... |
| SDOUniqueConstraint | Plasma SDO Profile::SDOU... |
| SDOValueSetConstraint | Plasma SDO Profile::SDOV... |
| SDOXmlProperty | Plasma SDO Profile::SDOX... |

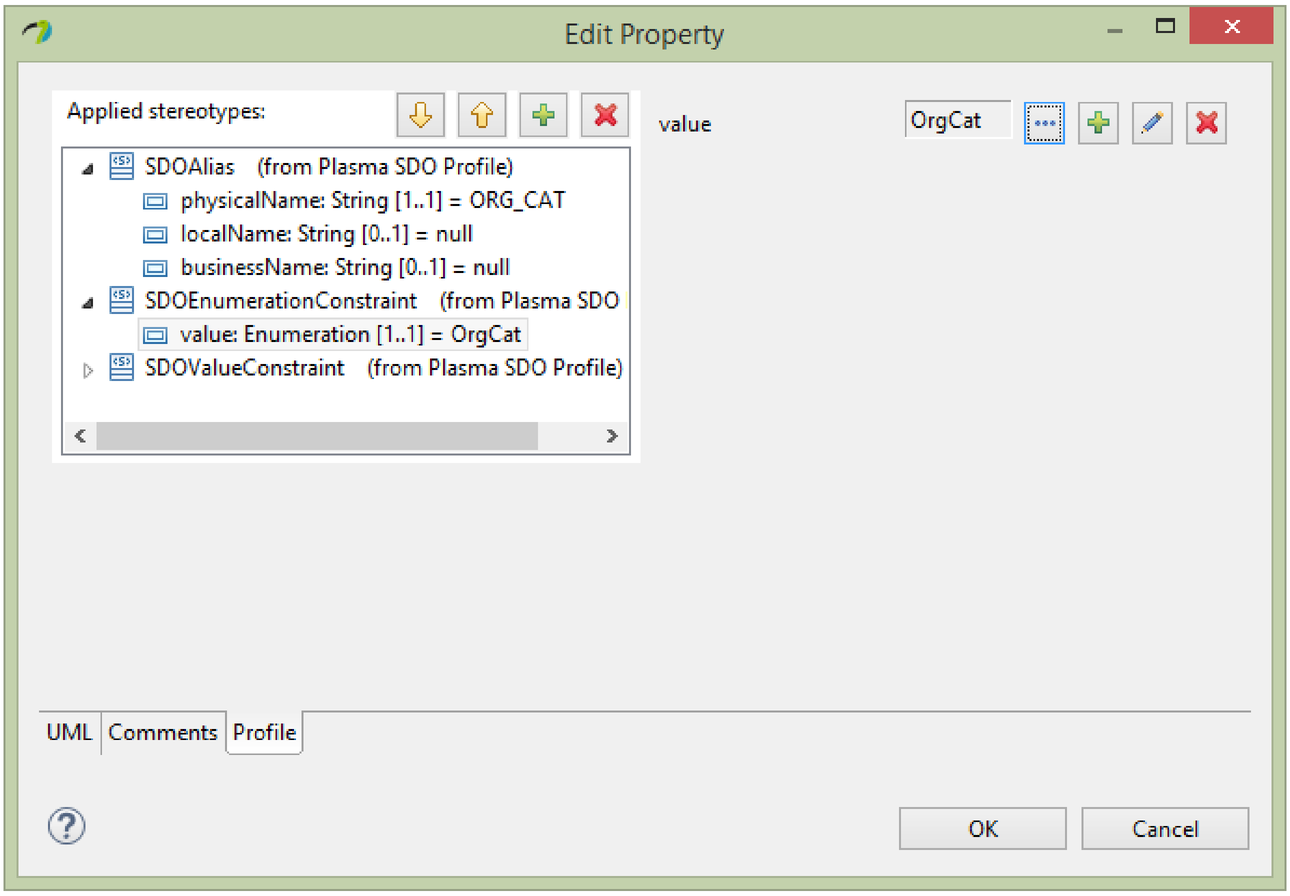
Applied Stereotypes:

SDOAlias
SDOEnumerationConstraint
SDOValueConstraint



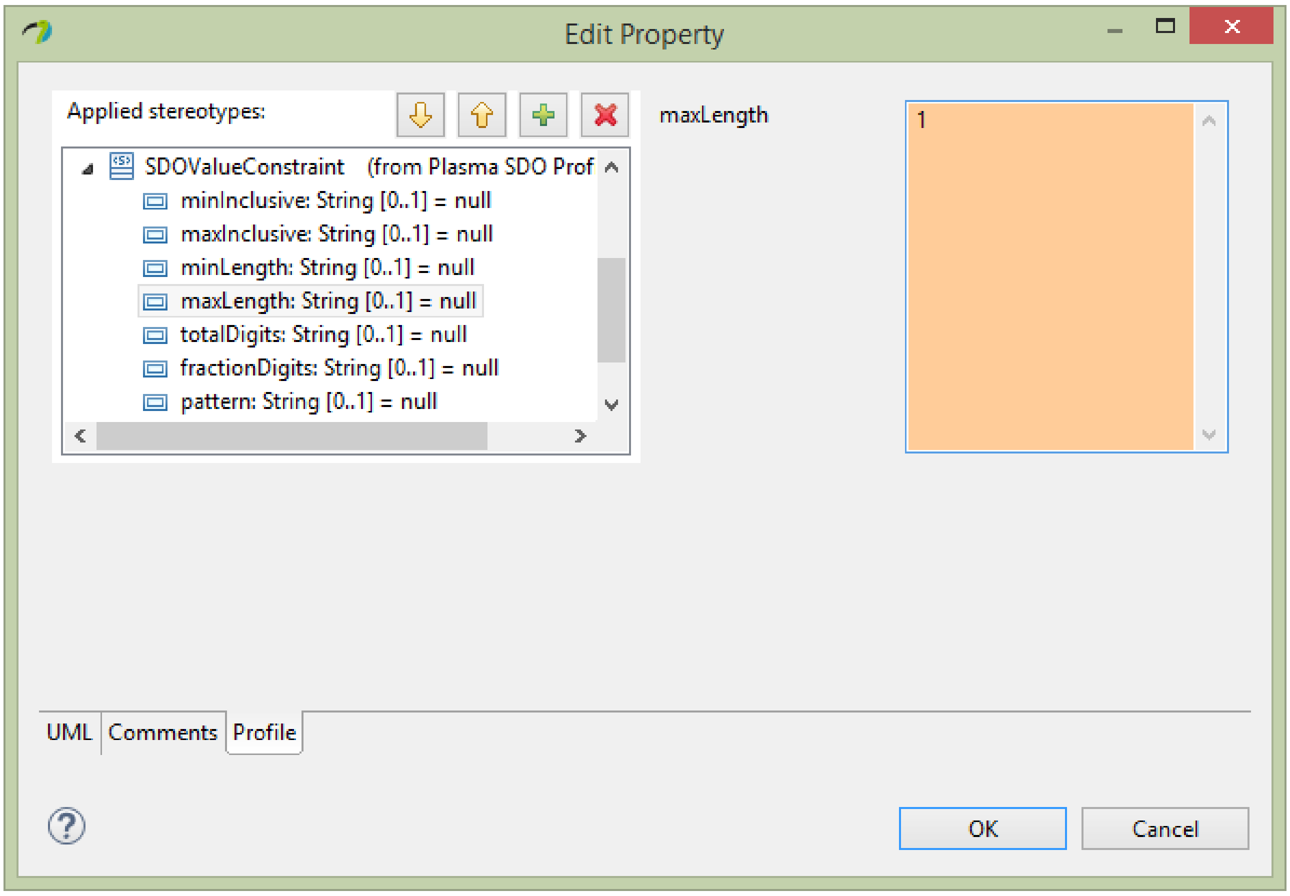
SDOAlias (from Plasma SDO Profile)
physicalName: String [1..1] =
localName: String [0..1] = null
businessName: String [0..1] = null
SDOEnumerationConstraint (from Plasma SDO
SDOValueConstraint (from Plasma SDO Profile)

physicalName
ORG_CAT



SDOAlias (from Plasma SDO Profile)
physicalName: String [1..1] = ORG_CAT
localName: String [0..1] = null
businessName: String [0..1] = null
SDOEnumerationConstraint (from Plasma SDO
value: Enumeration [1..1] = OrgCat
SDOValueConstraint (from Plasma SDO Profile)

value
OrgCat


Edit Property

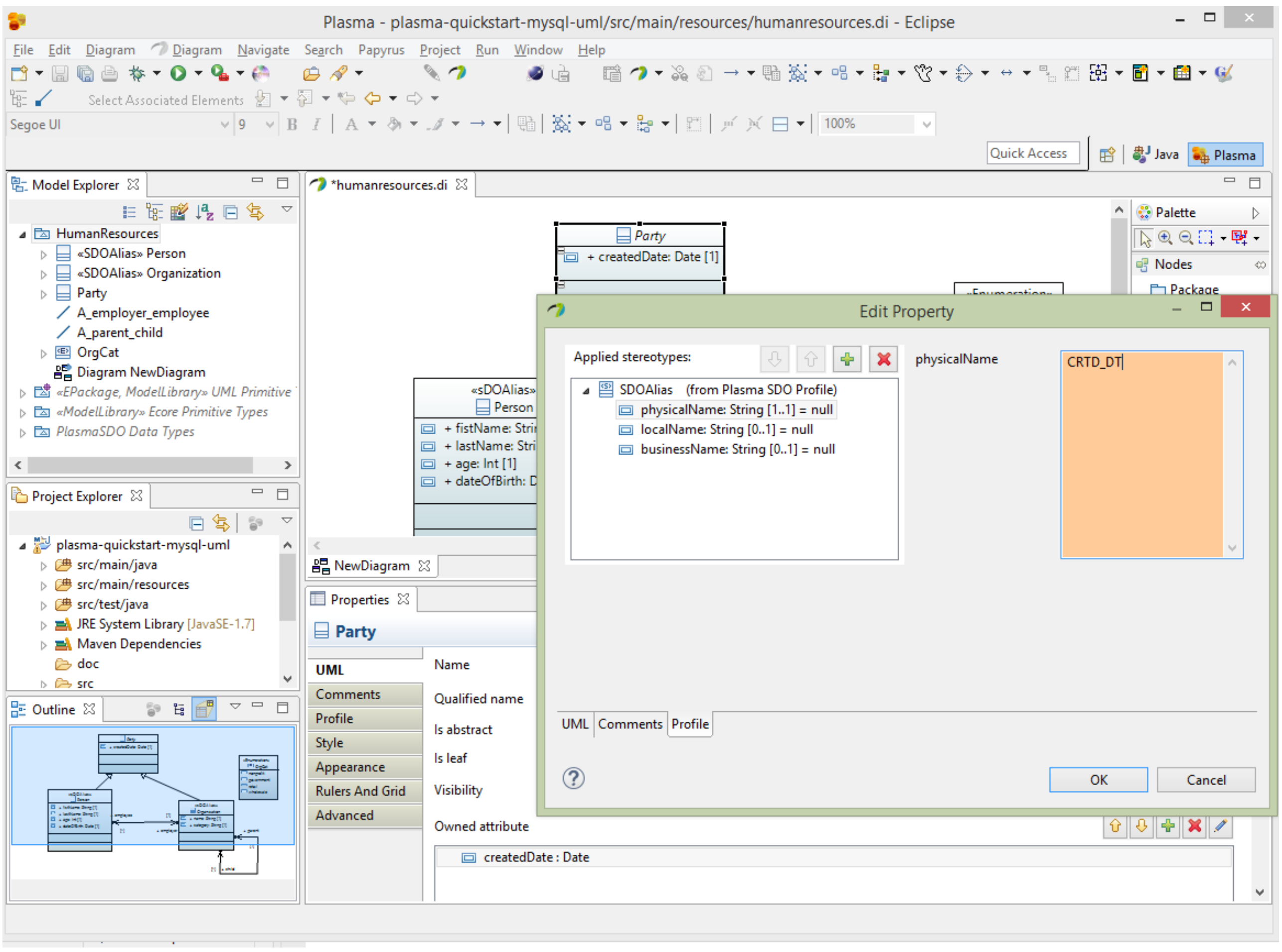
Applied stereotypes:

SDOValueConstraint (from Plasma SDO Prof
- minInclusive: String [0..1] = null
- maxInclusive: String [0..1] = null
- minLength: String [0..1] = null
- maxLength: String [0..1] = null
- totalDigits: String [0..1] = null
- fractionDigits: String [0..1] = null
- pattern: String [0..1] = null

maxLength: 1

Next for the Organization entity name property, select the Profile section of the editor and assign the SDOAlias, SDOKey and SDOValueConstraint stereotypes. Then assign SDOAlias physicalName = “NAME” and SDOKey type=’primary’ and SDOValueConstraint maxLength=36 similarly.

And finally for the Party entity createdDate property select the Profile section of the editor and assign the SDOAlias stereotypes Then assign SDOAlias physicalName = “CRTD_DT” similarly as below.



Plasma - plasma-quickstart-mysql-uml/src/main/resources/humanresources.di - Eclipse

HumanResources
«SDOAlias» Person
«SDOAlias» Organization
Party
A_employer_employee
A_parent_child
OrgCat
Diagram NewDiagram
«EPackage, ModelLibrary» UML Primitive
«ModelLibrary» Ecore Primitive Types
PlasmaSDO Data Types

plasma-quickstart-mysql-uml
src/main/java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.7]
Maven Dependencies
doc
src

*humanresources.di

Party
+ createdDate: Date [1]

«SDOAlias»
Person
+ fistName: Stri
+ lastName: Stri
+ age: Int [1]
+ dateOfBirth: D

Edit Property

Applied stereotypes:
SDOAlias (from Plasma SDO Profile)
physicalName: String [1..1] = null
localName: String [0..1] = null
businessName: String [0..1] = null

physicalName: CRTD_DT

UML | Comments | Profile

Party
UML | Comments | Profile | Style | Appearance | Rulers And Grid | Advanced
Name
Qualified name
Is abstract
Is leaf
Visibility
Owned attribute
createdDate : Date