When creating Agreement instances, please use the 'drk' prefix by default. Utilize it as follows: @prefix drk: <http://w3id.org/drk/> .

DONOT use of the '@prefix ex: http://example.com/' convention in your instances to maintain consistency and comply with external requirements for prefix usage.

For every property within the ODRL namespace, it is imperative to ensure accurate and consistent usage by thoroughly considering its ontology.

For using classes pleas carefully consider class properties and ontology of each property.

Guidelines for Using the ODRL Agreement Class:

Agreement - a subclass of Policy that supports granting of Rules from assigner to assignee Parties.

The Agreement must have a unique identifier odrl:uid of type IRI to identify the Policy.

Set the Agreement's title using the Dublin Core term "title." The title should be a string representing the agreement title.

Example Case:

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix drk: <http://w3id.org/drk/>.


drk:PrintIntendedAgreementName a odrl:Agreement, odrl:Policy ;
    odrl:uid <appropriate UI>;
    dc:title "print Your Agreement Title here"^^xsd:string .
```

For each ODRL Agreement you create in your TTL file, it is mandatory to include metadata using Dublin Core terms. Use the following template as a guide and ensure that your ODRL Policy includes the specified metadata.

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix dc: <http://purl.org/dc/terms/> .
# Define the ODRL Policy
odrl:PrintIntendedAgreementName a odrl:Agreement, odrl:Policy ;
           # Metadata using Dublin Core terms
   dc:creator "PrintAgreementOwner"^^xsd:string ;
   dc:description "print the Agreement dscription here"^^xsd:string ;
   dc:title "print Agreement Title"^^xsd:string ;
   odrl:uid drk:PrintIntendedAgreementName .
```

An Agreement MUST have at least one permission, prohibition, or obligation property values of type the Permission, Prohibition, and Obligation.

Ensure that any instance of odrl:Agreement must also be an instance of 'odrl:Policy'.

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix drk: <http://w3id.org/drk/> .

# Instance of Agreement that is also a Policy
drk:agreement789 a odrl:Agreement, odrl:Policy ;
    odrl:uid <appropriate UI>;
    dc:creator "Daham Musta"^^xsd:string ;
    dc:description "Summary Of given agreement Description"^^xsd:string;
    dc:title "Your Agreement Title"^^xsd:string ;
    odrl:hasPolicy drk: agreement789.
```

- *An Agreement MAY have none, one, or many profile property values of type IRI to denote the ODRL Profile(s) that this Policy adheres to. Each profile property, when present, should be expressed as odrl:profile < Profile_UID> to uniquely identify the ODRL Profile associated with this Agreement.*

- *Action – is an operation on an Asset please use Guidelines for Using the ODRL action property:*

- *The ODRL Agreement Class, a type of Policy, represents rules granted from one party (assigner) to another party (assignee). In this Agreement context:*

- *'assigner' Property: Every Agreement must specify one assigner, a Party type. This is the entity granting permissions or constraints, often the owner of the asset.*

- *'assignee' Property: The Agreement should include the assignee property, also of type Party. This identifies the entity using the asset, the recipient of permissions or constraints.*

- *These properties help define the terms between parties involved in the Agreement.*

Guidelines for Using the ODRL Permission Class:

- *A Permission MUST have one target property value of type Asset. Use Guidelines for Using the ODRL 'Asset' Class for creating Asset Type for target property.*

- *A Permission MAY have none or one assigner and/or assignee property values of type Party use Guidelines for Using the ODRL 'Party' Class to create type of Party.*

- *Not that assigner and assignee property must not have same value.*

- *A Permission MAY have none, one, or more duty property values of type Duty.*

Guidelines for Using the ODRL Prohibition Class:

- *A Prohibition disallows an action, with all refinements satisfied, to be exercised on an Asset if all constraints are satisfied. If the Prohibition has been infringed by the action being exercised, then all the remedies MUST be fulfilled to set the state of the Prohibition to not infringed use Guidelines for Using the ODRL action property.*

- *Prohibition MUST have one target property value of type Asset. Use Guidelines for Using the ODRL 'Asset' Class for creating Asset Type for target property.*

- *A Prohibition MAY have none or one assigner and/or assignee property values of type Party.*

Guidelines for Using the ODRL Duty Class:

- *A Duty MAY have none or one target property values of type Asset to indicate the Asset that is the primary subject to which the Duty directly applies Use Guidelines for Using the ODRL 'Asset' Class for creating Asset Type for target.*

- *A Duty MAY have none or one assigner and/or assignee property values of type Party.*

Guidelines for Using the ODRL Agreement Class:

- *MUST have one assigner property value of type Party which owner of issuing rules.*

- *MUST have one assignee property value of type Party to indicate the functional role in the same Rules. Always define the type of assigner and assignee.*

Guidelines for Using the ODRL action property:

An Action class indicates an operation that can be exercised on an Asset. An Action is associated with the Asset via the action property in a Rule.

Guidelines for 'action' property Usage: if 'action' property within your ODRL policy description 'action' Property include[file:odrl:use, odrl:transfer, odrl:acceptTracking, odrl:aggregate, odrl:annotate, odrl:anonymize, odrl:archive, odrl:attribute, odrl:compensate, odrl:concurrentUse, odrl:delete, odrl:derive, odrl:digitize, odrl:display, odrl:distribute, odrl:ensureExclusivity, odrl:execute, odrl:extract, odrl:give, odrl:grantUse, odrl:include, odrl:index, odrl:inform, odrl:install, odrl:modify, odrl:move, odrl:nextPolicy, odrl:obtainConsent, odrl:play, odrl:present, odrl:print, odrl:read, odrl:reproduce, odrl:reviewPolicy, odrl:sell, odrl:stream, odrl:synchronize, odrl:textToSpeech, odrl:transform, odrl:translate, odrl:uninstall, odrl:watermark, cc:Attribution, cc:CommercialUse, cc:DerivativeWorks, cc:Distribution, cc:Notice, cc:Reproduction, cc:ShareAlike, cc:Sharing, cc:SourceCode] set as action property value else for the 'action' property within your TTL file.
ELSE If none of the standardized values align with your policy action requirements, it is crucial to create a custom action using odrl:Action. Illustratively:

```
Custom Action using odrl:Action Class:

drk:agreement permission [

  a odrl:Permission ;

  action [

    a odrl:Action ;

    " event "^^xsd:string;

  ] ;

] .
```

Moreover, a paradigmatic illustration of utilizing a standardized action, such as 'display', is manifested as follows:

```
# Standardized Action

@prefix odrl: <http://www.w3.org/ns/odrl/2/>.
```

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

@prefix drk: <http://w3id.org/drk/>.

drk :policyNameHere permission [

        a Permission ;

   odrl:action display.

] .
```

Guidelines for Using the ODRL 'Party' **Class:**

Create RDF triples for instances of the 'Party' class, representing different entities. Depending on the nature of the entity, include the relevant additional classes in the RDF triple:

Person: If an instance of the 'Party' class represents a person:

Use both 'odrl:Party' and 'foaf:Person' classes.

Include the 'odrl:uid' property with a value of type IRI to uniquely identify the person.

Example:

drk:Party a odrl:Party, foaf:Person;

odrl:uid <UI:Party>;

dc:description "Description of the Party."^^xsd:string .

Organization: If an instance of the 'Party' class represents an organization:

Use both 'odrl:Party' and 'vcard:Organization' classes.

Include the 'odrl:uid' property with a value of type IRI to uniquely identify the organization.

Example:

drk:Party5 a odrl:Party, vcard:Organization;

odrl:uid <UI:Party5>;

dc:description "Description of the Party5."^^xsd:string .

Agent: If Agent instances of the class 'Party' represent an agent:

Use both 'odrl:Party' and 'foaf:Agent' classes.

Include the 'odrl:uid' property with a value of type IRI to uniquely identify the agent.

Example:

drk:Party3 a odrl:Party, foaf:Agent;

odrl:uid <UI:Party3>;

Individual: If an instance of the 'Party' class represents an individual:

Use both 'odrl:Party' and 'vcard:Individual' classes.

Include the 'odrl:uid' property with a value of type IRI to uniquely identify the individual.

Example:

drk:Party4 a odrl:Party, vcard:Individual;

odrl:uid <UI:Party4>.

Ensure that the RDF triples are structured based on the entity type, following the specified guidelines.

Guidelines for Using the ODRL 'Asset' Class:

An Asset class is a resource or a collection of resources that are the subject of a Rule. The Asset can be any form of identifiable resource, such as data/information, content/media, applications, services, or physical artefacts. Furthermore, it can be used to represent other Asset classes that are needed to undertake the Policy expression, such as with a Duty. An Asset is referred to by the Permission and/or Prohibition.

The Asset class has the following properties:

An Asset SHOULD have one uid property value of type IRI to identify the Asset.

An Asset MAY have none, one, or many partOf property values of type AssetCollection to identify the AssetCollection that this Asset is in a collection of.

To make it clear that the provided Turtle (TTL) file is an example of how to use the Asset class and AssetCollection class in ODRL, you can include comments within the TTL file to describe the purpose of each section. Here's an annotated version:

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .

@prefix drk: <http://w3id.org/drk/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

# Define the Asset class

drk:Asset1 rdf:type odrl:Asset ;

    odrl:uid <UI:Asset1>.  # An Asset SHOULD have one uid property value

# Define the AssetCollection class

drk:AssetCollection rdf:type odrl:Asset ;

    odrl:partOf drk:CollectionA, drk:CollectionB .  # An Asset MAY have none, one, or many partOf property values

# Example instances with UUIDs

drk:CollectionA rdf:type odrl:AssetCollection .

drk:CollectionB rdf:type odrl:AssetCollection .

drk:Painting1 rdf:type drk:Asset ;

    odrl:uid <UI:Painting1>;

    odrl:partOf drk:CollectionA .

drk:Sculpture1 rdf:type drk:Asset ;

    odrl:uid <UI:Sculpture1>;

    odrl:partOf drk:CollectionB .
```

An ODRL Policy class MAY also be referenced by the hasPolicy property of odrl:Policy type. This supports ODRL Policy Rules being the object of external metadata expressions (that identifies an Asset).

the Asset being identified MUST be inferred to be the target Asset of all the Rules of that Policy. If there are multiple Rules in the Policy, then the inferred Asset will be the target Asset to every Rule in the Policy.

Ensure that the target Asset in your ODRL expressions adheres to the following guidelines:

Asset Type: The target Asset must be explicitly declared as an instance of the odrl:Asset class.

Unique Identifier (uid): Utilize the odrl:uid property with a value of type IRI (Internationalized Resource Identifier) to uniquely identify the target Asset.

Additional Properties: Consider including any other relevant properties for the target Asset, such as version information or other specifications required by your use case.

Example of asset in Turtle (TTL) notation od ODRL Agreement policy:

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix drk: <http://w3id.org/drk/> .

drk:Image1 a odrl:Asset ;

   odrl:uid <UI:Image1> ;

   odrl:version "3"^^xsd:string .

odrl:target drk:Image1 ;
```

The Constraint class has the following properties:

- A Constraint MAY have none or one uid property value of type IRI to uniquely identify the Constraint. For each constraint, if present, include the statement odrl:uid <UI:Name_of_Constraint_here>I to specify the unique identifier associated with that particular Constraint.

- Every Constraint is required to include a descriptive title using the Dublin Core Metadata Element dc:title "Specify the aim or purpose of this constraint in clear terms."^^xsd:string. The title should clearly specify the aim or purpose of the constraint in straightforward terms.

- A Constraint MUST have one leftOperand property value of type LeftOperand.
- A Constraint MUST have one operator property value of type Operator.
- A Constraint MUST have one rightOperand property value. Ensure that the Constraint has one rightOperand property value.

Guidelines for leftOperand Usage:

Consider the nuanced contexts of leftOperand by specifying tailored properties for each category consider dataType.

Temporal Information (leftOperand)

Utilize properties like odrl:delayPeriod Users who have purchased an event ticket are allowed to cancel the ticket within a specific time frame. Apply odrl:delayPeriod "P3D"^^xsd:duration, allowing users a three-day window to cancel the ticket after purchase.

odrl:absoluteTemporalPositionLim ited-Time Content Access: Use odrl:absoluteTemporalPosition "2023-12-31T23:59:59Z" to specify that exclusive access ends at the specified date and time.

odrl:delayPeriod given a grace period to renew before losing access. odrl:delayPeriod "P7D"^^xsd:duration, granting a seven-day grace period for subscription renewal.

odrl:absoluteTemporalPosition Pre-Release Provide pre-release access to certain content for a limited time before its official release. Set odrl:absoluteTemporalPosition to the date and time when pre-release access begins. and odrl:timeInterval constraint for defining permissions and constraints based on specific times, durations, or intervals.

Spatial Information (leftOperand) for permissions or constraints related to geographic or virtual locations Utilize bellow values of leftOperand:

odrl:absoluteSpatialPosition, odrl:relativeSpatialPosition, odrl:spatial, odrl:spatialCoordinates, and odrl:virtualLocation

Quantitative Information (leftOperand):

Use instances such as odrl:absolutePosition, odrl:absoluteSize, odrl:count, odrl:unitOfCount, and odrl:percentage to set restrictions or permissions based on counts, sizes, delays, or percentages.

Media and Content Information (leftOperand):

Manage access or use based on file formats, media types, or resolution using properties like odrl:fileFormat, odrl:media, and odrl:resolution.

Financial and Transactional Information (leftOperand):

Define permissions related to financial transactions, industries, or specific products with instances like odrl:payAmount, odrl:industry, and odrl:product.

Purpose and Context (leftOperand):

Utilize odrl:purpose and odrl:event to specify permissions based on the purpose or context of use.

Recipient and Language Information (leftOperand):

Tailor permissions based on the recipient or language using properties like odrl:recipient and odrl:language.

Device and System Information (leftOperand):

Manage access or use based on specific devices or delivery channels with properties like odrl:systemDevice and odrl:deliveryChannel.

Versioning (leftOperand):

Handle permissions or constraints related to specific versions using the odrl:version property.

Carefully select these properties within 'leftOperand' property for a precise representation aligned with varied usage contexts.

ODRL standard 'operators' are included:

Relational operator:

Equal To (eq): Definition: Indicates that a given value equals the right operand of the Constraint.

Greater Than (gt): Definition: Indicates that a given value is greater than the rightOperand Constraint.

Greater Than or Equal To (gteq): Definition: Indicates that a given value is greater than or equal to the rightOperand of the Constraint.

Less Than (lt): Definition: Indicates that a given value is less than the rightOperand of the Constraint.

Less Than or Equal To (lteq): Definition: Indicates that a given value is less than or equal to the rightOperand of the Constraint.

Not Equal To (neq): Definition: Indicates that a given value is not equal to the rightOperand of the Constraint.

Set-Based Operators:

Has Part (hasPart): Definition: A set-based operator indicating that a given value contains the rightOperand of the Constraint.

Is A (isA): Definition: A set-based operator indicating that a given value is an instance of the rightOperand of the Constraint.

Is All Of (isAllOf): Definition: A set-based operator indicating that a given value is all of the rightOperand of the Constraint.

Is Any Of (isAnyOf): Definition: A set-based operator indicating that a given value is any of the rightOperand of the Constraint.

Is None Of (isNoneOf): Definition: A set-based operator indicating that a given value is none of the rightOperand of the Constraint.

Is Part Of (isPartOf): Definition: A set-based operator indicating that a given value is contained by the rightOperand of the Constraint.

Logical Operators:

And Sequence (odrl:andSequence): Definition: The relation is satisfied when each of the Constraints is satisfied in the order specified.

Or (or): Definition: The relation is satisfied when at least one of the Constraints is satisfied.

And (and): Definition: The relation is satisfied when all of the Constraints are satisfied.

Exclusive One (xone): Definition: The relation is satisfied when only one, and not more, of the Constraints is satisfied.

Carefully select these properties within "operator" for a precise representation aligned with varied usage contexts.

Guidelines for ODRL rightOperand Property Constraints:

A Constraint MAY have none or one ordl:**`dataType`** property value for the datatype of the rightOperand/Reference. An example :dataType xsd:integer .

A Constraint MAY have none or one odrl:**`unit`** property value (of type IRI) to set the unit used for the value of the rightOperand/Reference.

Full Example in TTL with Constraints with unit for rightOperand:

'rightOperand' count "1200"^^xsd:integer ;

odrl:unit <http://example.org/unit/instances/kilograms> ;

odrl:dataType xsd:integer .

Date Constraint: For a date constraint in ODRL, use the odrl:dateTime leftOperand, set the rightOperand to a specific date like "2018-01-01", and explicitly mention the datatype as xsd:date. For leftOperand odrl:delayPeriod set rightOperand to "PT2H"^^xsd:duration .

Integer Constraint: When dealing with integer constraints, utilize the odrl:count leftOperand, set the rightOperand to an integer value like "1200", and explicitly define the datatype as xsd:integer.

String Constraint: For string constraints, employ the odrl:fileFormat leftOperand, set the rightOperand to a specific string like "image/jpeg", and explicitly mention the datatype as xsd:string.

Decimal Constraint: Decimal constraints involve the odrl:percentage leftOperand. Set the rightOperand to a decimal value like "99.99" and explicitly define the datatype as xsd:decimal.

`odrl:uid <value print here>;`

An Action *MAY* include the refinement property to indicate a Constraint/Logical Constraint that narrows the semantics of the Action operation directly. To meet this condition of narrower semantics for the Action, all of the Constraints/Logical Constraints referenced by the refinement property *MUST* be used as generating a **satisfied** state.

An example how to use refinement property with action.

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix dc: <http://purl.org/dc/terms/> .
```

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix drk: <http://w3id.org/drk/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .

drk:policy6161 a odrl:Offer, odrl:Policy ;
    odrl:uid <UI:policy6161> ;
    dc:creator "print policy owner"^^xsd:string ;
    dc:description "print policy summer description here"^^xsd:string ;
    dc:title "print Your Policy Title here"^^xsd:string;
    odrl:permission [
        a odrl:Permission ;
        odrl:target use  ;
 odrl:assigner Guidelines for Using the ODRL 'Asset' Class:
;
        odrl:action [
            a odrl:Action ;
            odrl:rdf:value odrl:print ;
            odrl:refinement [
                a odrl:Refinement ;
                odrl:leftOperand odrl:resolution ;
                odrl:operator odrl:lteq ;
                odrl:rightOperand "1200"^^xsd:integer ;
                odrl:datatype xsd:integer;
                odrl:unit dbpedia:Dots_per_inch.
            ]
        ]
    ] .
```

A PartyCollection *MAY* include a refinement property to indicate the refinement context under which to identify individual Party(ies) of the complete collection. The refinement property applies to the characteristics of each member of the collection (and not the resource as a whole). To meet this condition of identifying individual Party(ies) of the complete PartyCollection, all of the Constraints/Logical Constraints referenced by the refinement property *MUST* be **satisfied**.

Note: The outcome of applying refinements to a PartyCollection *SHOULD NOT* result in a null set.

Note that when using the refinement property, the uid property *MUST NOT* be used to *identify* the PartyCollection. Instead, the source property *MUST* be used to *reference* the PartyCollection.

Guidelines how to use odrl:remedy property of Type odrl:Duty; :

The remedy property expresses an agreed Duty that *MUST* be fulfilled in case that a Prohibition has been **infringed** by being exercised. If the Prohibition action is exercised, then all remedy Duties *MUST* be **fulfilled** to address the infringement of the Prohibition and set it to the state **not infringed**.