Complex Changes are represented a json-formatted files (one per change). Each json file has the following format:

{

"Complex\_Change": "name of the change",

"Priority": 1,

"Complex\_Change\_Parameters": [

{

"cc\_param1": "sc1:-subclass"

}

],

"Simple\_Changes": [

{

"Simple\_Change": "ADD\_SUPERCLASS",

"Simple\_Change\_Uri": "sc1",

"Is\_Optional": false,

"Selection\_Filter": ["sc1:-superclass =rdf:Class "],

"Mapping\_Filter": "",

"Join\_Filter": ""

}

],

"Version\_Filters": [

{

"Subject": "sc1:ADD\_SUPERCLASS:-subject",

"Predicate": "rdfs:subClassOf",

"Object": "rdfs:Resource",

"Presence": "EXISTS\_IN\_V2"

}

]

}

where

* **Complex\_Change:** the name of the complex change.
* **Priority:** its priority which can be any double number.
* **Complex\_Change\_Parameters:** the parameter names of the complex change along with the simple change parameter names which are binded with.
* **Simple\_Changes:** an array of simple changes which consist the complex change. For each simple change we have the following fields:
  + **Simple\_Change:** the name of the simple change.
  + **Simple\_Change\_Uri:** an identifier of the simple change which can be used to separate simple changes of the same name.
  + **Is\_Optional:** a Boolean variable which indicates whether the simple change is optional (true) or mandatory (false).
  + **Selection\_Filter:** an array of filters which assigns selective values upon simple change parameters. Can be empty as well.
  + **Mapping\_Filter:** an array of filters which express mappings upon simple change parameters. Can be empty as well.
  + **Join\_Filter:** an array of filters which express joins across different parameters of simple changes. Can be empty as well.
* **Version\_Filters:** an array of version filters which have to be satisfied. Each version filter is essentially a triple which have to exist (or not) in either the new or old dataset version. Each part of a triple (subject, predicate, object) can be a) a specific uri, b) a complex change parameter, c) a simple change parameter. The presence of the triple is denoted by a flag which takes one of the values namely, EXISTS\_IN\_V2, EXISTS\_IN\_V1, NOT\_EXISTS\_IN\_V2, NOT\_EXISTS\_IN\_V1.

**Simple Changes Used for IdeaGarden Complex Changes**

|  |  |  |
| --- | --- | --- |
| Change | **Add\_Type\_To\_Individual(a, b)** | **Delete\_Type\_From\_Individual(a, b)** |
| Intuition | Type b of individual a is added | Type b of individual a is deleted |
| Parameters | a = the individual  b = the new type | a = the individual  b = the new type |
| Added Triples | <a, rdf:type, b> | - |
| Deleted Triples | - | <a, rdf:type, b> |

|  |  |  |
| --- | --- | --- |
| Change | **Add\_Property\_Instance(a1,b,a2)** | **Delete\_Property\_Instance(a1,b,a2)** |
| Intuition | Add property instance of property b | Delete property instance of property b |
| Parameters | a1 = The subject  b = The property  a2 = The object | a1 = The subject  b = The property  a2 = The object |
| Added Triples | <a1, b, a2> | - |
| Deleted Triples | - | <a1, b, a2> |