

**IBM Software**

# **RELM View Development**

**How to develop RELM View ?**

## View Development Steps

1. Write a mock-up of View you want to see
2. Investigate the data are accessible by LQE
3. Investigate predicate for SPARQL query
4. Write SPARQL
5. Execute SPARQL
6. Fix Container
7. Modify Node/UI Type
8. Write Connection
9. Execute View
10. Consider the usage of Jazz Reporting Service (JRS) ( v5.0.2 or later)



# View Development Steps - Investigate the data are accessible by LQE

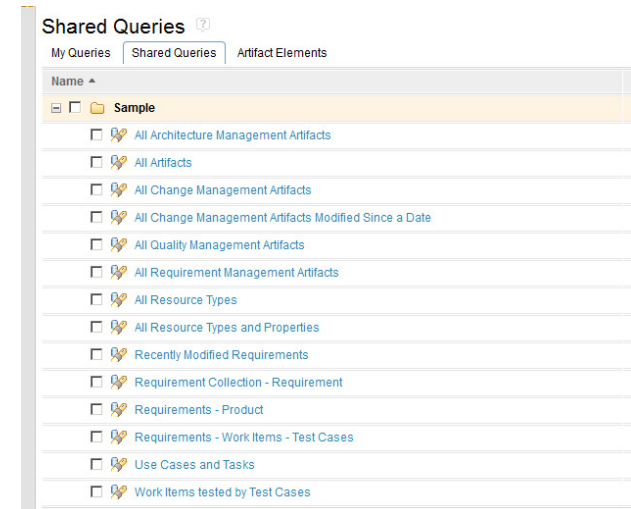


- Investigate necessary data can be obtained by LQE
  - Try Query > Shared Queries > Sample Query
    - For RTC Workitems, " All Change Management Artifacts" is useful. For other artifacts, the corresponding Queries are available too.
    - By default, all results are returned. Would be better if you filtered data.
    - Example: Specify ID. ( ID=183 will be found. )

```

PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX osc: <http://open-services.net/ns/core#>
PREFIX osc_cm: <http://open-services.net/ns/cm#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?resource ?id ?shortTitle ?title ?modified ?type
WHERE {
    ?resource a osc_cm:ChangeRequest;
               dcterms:identifier "183"^^xsd:string.
    OPTIONAL {
        ?resource
            dcterms:title ?title ;
            dcterms:type ?type ;
            osc:shortTitle ?shortTitle ;
            dcterms:modified ?modified .
    }
}
ORDER BY ASC(?type) ASC(?id)

```



## View Development Steps - Investigate the data are accessible by LQE (Cont.)

SPARQL

- Example: Specify a part of a title. Will find Worktiem of which title has "leak"

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX osc: <http://open-services.net/ns/core#>
PREFIX osc_cm: <http://open-services.net/ns/cm#>
SELECT ?resource ?id ?shortTitle ?title ?modified ?type
WHERE {
  ?resource a osc_cm:ChangeRequest .
  OPTIONAL {
    ?resource
      dcterms:identifier ?id ;
      dcterms:title ?title ;
      dcterms:type ?type ;
      osc:shortTitle ?shortTitle ;
      dcterms:modified ?modified .
  }
  FILTER (CONTAINS(str(?title), "leak"))
}
ORDER BY ASC(?type) ASC(?id)
```

- -> Update a sample query anyway to find out the artifact( RTC WI ) you want to visualize.

## View Development Steps - Investigate the data are accessible by LQE (Cont.)

- Left click > "Show Properties" from RELM Query Result,
  - Each line has a data you can access by LQE
    - Easy way to see if the necessary data is indexed
    - This action is often used. It is suggested to keep the generated SPARL Queries.

Queries > Shared Queries >

### Query Result : All Change Management Artifacts

Show All Items per Page

#	ID	Title
1		Change Request 183: SysCR: Determine impact of change to leak ind

- Open Artifact
- Show Properties
- Start Analysis
- Open View...



#	Property	Value
1	Type	cmfChangeRequest
2	Description	U.S. regulations now require all handheld meter readers to provide both visual indication as well as audible alarms when leaks are detected. Analysis needs to be completed to determine how this change impacts system requirements, hardware, software and product variants.
3	Title	SysCR: Determine impact of change to leak indicator requirements in regards to U.S. regulations
4	Type	change_request
5	TeamArea	_70_7L3FXEeKPMDrWkualg
6	ModifiedBy	rational
7	ProgressTracking	progressTracking
8	State	ChangeRequest.state.s3
9	TimeSheet	rtc_cm.timeSheet
10	Subscribers	rational
11	Com.ibm.team.workitem.linktype.parentworkitem.parent	185: Implement leak indicator
12	PlannedFor	__8rPGWFXEeKPMDrWkualg
13	Repository	repository
14	Com.ibm.team.workitem.linktype.parentworkitem.children	186: System Task
15	Com.ibm.team.workitem.linktype.blocksworkitem.blocks	184: RCR: Change requirements for leak indicator requirements in regards to U.S. regulations

## View Development Steps - Investigate the data are accessible by LQE (Cont.)

### • RELM Query Result

- Black characters (No link is displayed in a hover window when the mouse is over the characters.) : A text is returned.
  - This can be displayed on RELM View.
- Blue characters ( Link is displayed in the hover window when the mouse is over the characters.) :URL is actually returned and RELM extracts the human readable characters from the data source tool ( for example, RTC ) and displays them instead.
  - Need more investigation to see if they can be displayed on RELM View. (You can expect if the human readable characters are displayed here, they can be also displayed in RELM View. )
  - Understand the difference between these two strings

<https://ssejtsserver:9443/ccm/resource/itemName/com.ibm.team.workitem.Worl>

Show All ▾ Items per Page Previous | 1 - 43 of 43 |

#	Property	Value
1	Type	cm#ChangeRequest
2	Description	U.S. regulations now require all handheld meter readers to provide determine how this change impacts system requirements, hardwar
3	Title	SysCR: Determine impact of change to leak indicator requirements
4	Type	change_request
5	TeamArea	_70_7L3FXEeKPYPMDrWKualg
6	ModifiedBy	rational
7	ProgressTracking	progressTracking <a href="https://ssejtsserver:9443/jts/users/rational">https://ssejtsserver:9443/jts/users/rational</a>
8	State	ChangeRequest.state.s3

A text is returned for "Description"

URI is returned for "ModifiedBy" ( a text in the hover window ) -> Need to investigate it more.

## View Development Steps - Investigate the data are accessible by LQE (Cont.)

- Further investigation to see the data can be displayed. - Can be displayed
  - Do Show Properties for the blue characters
    - Example: Left click > "Show Properties" to "ModifiedBy" value.
    - Verify that necessary data (for example, "Name" value) can be obtained.
  - In the right example, WI's "ModifiedBy" value ("Name") can be accessible through by SPARQL. (You can see "rational")



#	Property	Value
1	Type	Person
2	Name	rational
3	Nick	rational
4	Mbox	mailto:rational@us.ibm.com
5	Archived	false



## View Development Steps - Investigate the data are accessible by LQE (Cont.)

- Further investigation to see the data can be displayed. - Can't be displayed
  - Do Show Properties for the blue characters
    - Example: Left click > "Show Properties" to "PlannedFor".
    - Nothing is displayed.
    - -> LQE doesn't index it.
  - In the right example, WI's "PlannedFor" value can't be accessible through by SPARQL. (You can't see "Release 1.0", for example. )
    - Anything related to RTC process aren't indexed for now. (<https://jazz.net/jazz/web/projects/Jazz%20Foundation#action=com.ibm.team.workitem.viewWorkItem&id=329247>)
    - In another word, the data you can obtain is only URI for "PlannedFor". If you need URI only, there is no issue.

8	state	ChangeRequest.state.ss
9	TimeSheet	rtc_cm.timeSheet
10	Subscribers	rational
11	Com.ibm.team.workitem.linktype.parentworkitem.parent	185: Implement leak indicator
12	PlannedFor	<a href="https://ssejtssserver:9443/ccm/oslc/iterations/_6rPGwXFXEeKPYMdrWKualg">_6rPGwXFXEeKPYMdrWKualg</a>
13	Repository	repository
14	Com.ibm.team.workitem.linktype.parentworkitem.children	198: System Task



Queries >

[https://ssejtssserver:9443/ccm/oslc/iterations/\\_6rPGwXFXEeKPYMdrWKualg](https://ssejtssserver:9443/ccm/oslc/iterations/_6rPGwXFXEeKPYMdrWKualg)

Show All ▾ Items per Page ◀ Previous | 0 - 0 of 0 | Next ▶

#	Property
---	----------

## View Development Steps - Investigate the data are accessible by LQE (Cont.)

- If you need to do SPARQL query to find out data which aren't indexed by LQE ?
  - Open PMR to submit RFE against a tool to let LQE index its data. ( In some cases, LQE intentionally doesn't index data. Ask Rational Customer Support to decide this request is APAR or RFE. ) (Note: You can't predict when this will be fixed or enhanced.)
  - If you need to query such data now, you need to obtain RDF for the given URI by accessing it as REST api, and add the returned RDF as LQE "Vocabularies" ( But it isn't recommended. )
    - If you can't get RDF, there is no way. (Example is ChangeSet)
  - Example: You may get RDF by using OSLC REST API. (Firefox's Rest Client is useful. See appendix page.)

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:rtc_cm="http://jazz.net/xmlns/prod/jazz/rtc/cm/1.0/" >
<rdf:Description rdf:about="https://ssejtssserver:9443/ccm/oslc/iterations/_6rPGwXFXEeKPYMdrWKualg">
<rdf:type rdf:resource="http://jazz.net/xmlns/prod/jazz/rtc/cm/1.0/iteration"/>
<rtc_cm:projectArea rdf:resource="https://ssejtssserver:9443/ccm/oslc/projectareas/_6ngdX3FXEeKPYMdrWKualg"/>
<rtc_cm:timeline rdf:resource="https://ssejtssserver:9443/ccm/oslc/timelines/_6rPGwHFXEeKPYMdrWKualg"/>
<rtc_cm:archived rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">false</rtc_cm:archived>
<rtc_cm:hasDeliverable rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</rtc_cm:hasDeliverable>
<dcterms:description rdf:parseType="Literal">
</dcterms:description>
<dcterms:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Release 1.0</dcterms:title>
<dcterms:identifier rdf:datatype="http://www.w3.org/2001/XMLSchema#string">R1</dcterms:identifier>
</rdf:Description>
</rdf:RDF>
```

The screenshot shows the Rational interface with a blue arrow pointing to the 'Add Vocabulary' button. To the right, a table displays the properties of an iteration:

#	Property	Value
1	Type	iteration
2	Description	
3	Title	Release 1.0
4	Identifier	R1
5	HasDeliverable	true
6	Archived	false
7	Timeline	_6rPGwHFXEeKPYMdrWKualg
8	ProjectArea	_6ngdX3FXEeKPYMdrWKualg

The screenshot shows a dialog box titled 'Load vocabulary from URL'. It has two radio buttons: 'Load vocabulary from URL' (which is unselected) and 'Upload vocabulary document' (which is selected). The 'Upload vocabulary document' option has a 'Browse...' button and the filename 'PlannedFor.txt'. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'. A blue arrow points from the 'Vocabularies' menu item in the left sidebar to the dialog box.

## View Development Steps - Investigate predicate for SPARQL Query

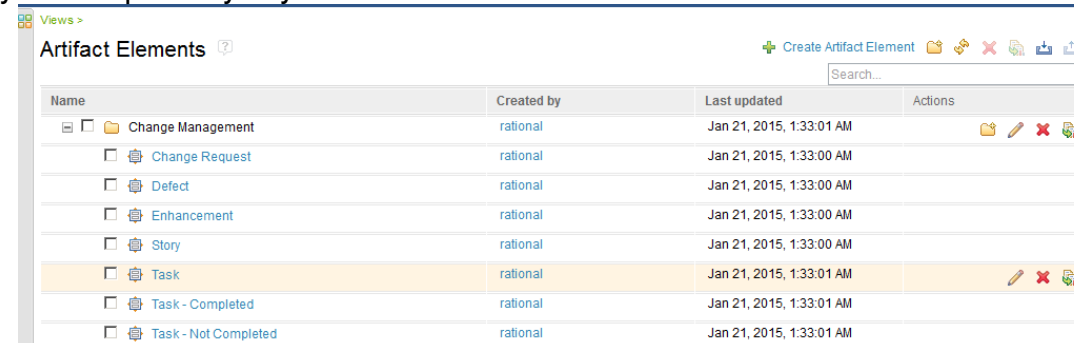
- "predicate" is necessary for SPARQL to query data ( for example, title )
  - From RELM Query result page, move the mouse cursor on Property so that you will see URI in the hover windows. This URI is predicate value.
  - Example: In below, predicate to find WI's children is "  
<http://jazz.net/xmlns/prod/jazz/rtc/cm/1.0/com.ibm.team.workitem.linktype.parentworkitem.children>"
  - You can use Browser's "Copy Link Location" to copy URI value.
  - In most cases, not a raw URI value but PREFIX + keyword is written in SPARQL
  - Example:  
 PREFIX rtc\_cm: <<http://jazz.net/xmlns/prod/jazz/rtc/cm/1.0/>>  
 rtc\_cm:com.ibm.team.workitem.linktype.parentworkitem.children ?target.

12	Parent of	<a href="#">_on SWA ALER TmduwKualg</a>
13	Repository	<a href="#">repository</a>
14	<a href="#">Com.ibm.team.workitem.linktype.parentworkitem.children</a>	186: System Task
15	<a href="#">Com.ibm</a>	<a href="#">ments for leak indicator requirements in regards to U.S. regulations</a>
16	FiledAgainst	<a href="#">_JONLSNP ALER TmduwKualg</a>
17	ResolvedBy	<a href="#">unassigned</a>

## View Development Steps - Write SPARQL

SPARQL

- Once it is clear that the data can be obtained by "Show Properties", and the necessary predicate value is obvious, you can start SPARQL development
  - Start SPARQL development to find out the data ( title, owner, due ) to be displayed in RELM View.
    - SPARQL development is a start point for RELM View development.
    - Over 60% time for RELM View development is spent for SPARQL development.
    - If you don't need View, SPARQL Query result might be what your customer wants.
  - It is suggested to keep any SPARQL to be developed for the investigation in View > My Queries.
    - Run Query in My Queries -> Copy it to View configuration - You may repeat these steps.
- As for SPARQL, you need to refer to spec(<http://www.w3.org/TR/sparql11-query/>)
  - If you don't know how to access the value for each artifacts (for example, CM or RM. Product Definition might be hard to find out how to access its data), you can try to see given Artifact Element(AE) sources. They have AE specific notations (for example, \$if(xyz)\$), but they are helpful anyway.
  - If you still don't know, you can open PMR.



Name	Created by	Last updated	Actions
Change Management	rational	Jan 21, 2015, 1:33:01 AM	
Change Request	rational	Jan 21, 2015, 1:33:00 AM	
Defect	rational	Jan 21, 2015, 1:33:00 AM	
Enhancement	rational	Jan 21, 2015, 1:33:00 AM	
Story	rational	Jan 21, 2015, 1:33:00 AM	
Task	rational	Jan 21, 2015, 1:33:01 AM	
Task - Completed	rational	Jan 21, 2015, 1:33:01 AM	
Task - Not Completed	rational	Jan 21, 2015, 1:33:01 AM	

## View Development Steps - Write SPARQL (Cont.)

SPARQL

- SPARQL Query sample

- Find an artifact of which type is "ActionItem" and of which Tag has "top", with its State, its children / grandchildren with their State. and its grandchildren's ChangeSet

- #!Label row has a definition of the column labels. The order of variables in Projection must be same as these lines' order. (if there is no definition, the variable names are displayed.)

```

#!Label L1 WI
#!Label L1 State
#!Label L2 WI
#!Label L2 State
#!Label L3 WI
#!Label L3 State
#!Label L3 Change Sets
#!Format ?{resourceL1}
#!Format ?{stateL1}
#!Format ?{resourceL2}
#!Format ?{stateL2}
#!Format ?{resourceL3}
#!Format ?{stateL3}
#!Format ?{changeSets}

PREFIX osc_cm: <http://open-services.net/ns/cm#>
PREFIX osc_cm_x: <http://open-services.net/ns/cm-x#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rtc_cm: <http://jazz.net/xmlns/prod/jazz/rtc/cm/1.0/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select ?resourceL1 ?stateL1 ?resourceL2 ?stateL2 ?resourceL3 ?stateL3 ?changeSets where {
#select * where {
# L1 -> L2
?resourceL1
dcterms:subject "top"^^xsd:string;
dcterms:type "ActionItem"^^xsd:string;
osc_cm:status ?stateL1.
OPTIONAL { ?resourceL1 rtc_cm:com.ibm.team.workitem.linktype.parentworkitem.children ?resourceL2.
BIND(if(bound(?resourceL2), ?resourceL2, rdf:nil) as ?resourceL2_)

#L2 -> L3
OPTIONAL { ?resourceL2_ osc_cm:status ?stateL2. }
OPTIONAL { ?resourceL2_ rtc_cm:com.ibm.team.workitem.linktype.parentworkitem.children ?resourceL3.
BIND(if(bound(?resourceL3), ?resourceL3, rdf:nil) as ?resourceL3_)

#L3 -> ...
OPTIONAL { ?resourceL3_ osc_cm:status ?stateL3. }
OPTIONAL { ?resourceL3_
rtc_cm:com.ibm.team.filesystem.workitems.change_set.com.ibm.team.scm.ChangeSet ?changeSets.
}

```

## View Development Steps - Execute SPARQL

- Run SPARQL in RELM Query

- Repeat to run SPARQL in Query > My Queries to verify its result
- Remember two types of characters ( black characters ( a text is returned ) and blue characters ( an URI is returned. )
- You can download the result as CSV. -> You can process it by Excel
  - Black characters -> They are in CSV.
  - Blue characters -> URIs are in CSV. (In example below, note that RELM Query shows the readable string instead for blue characters.) You need to add SPARQL if you want to put such a readable string in CSV.

Queries > My Queries >

### Query Result : CMQueryTest

Show All  Items per Page

« Previous | 1 - 6 of 6 | Next »

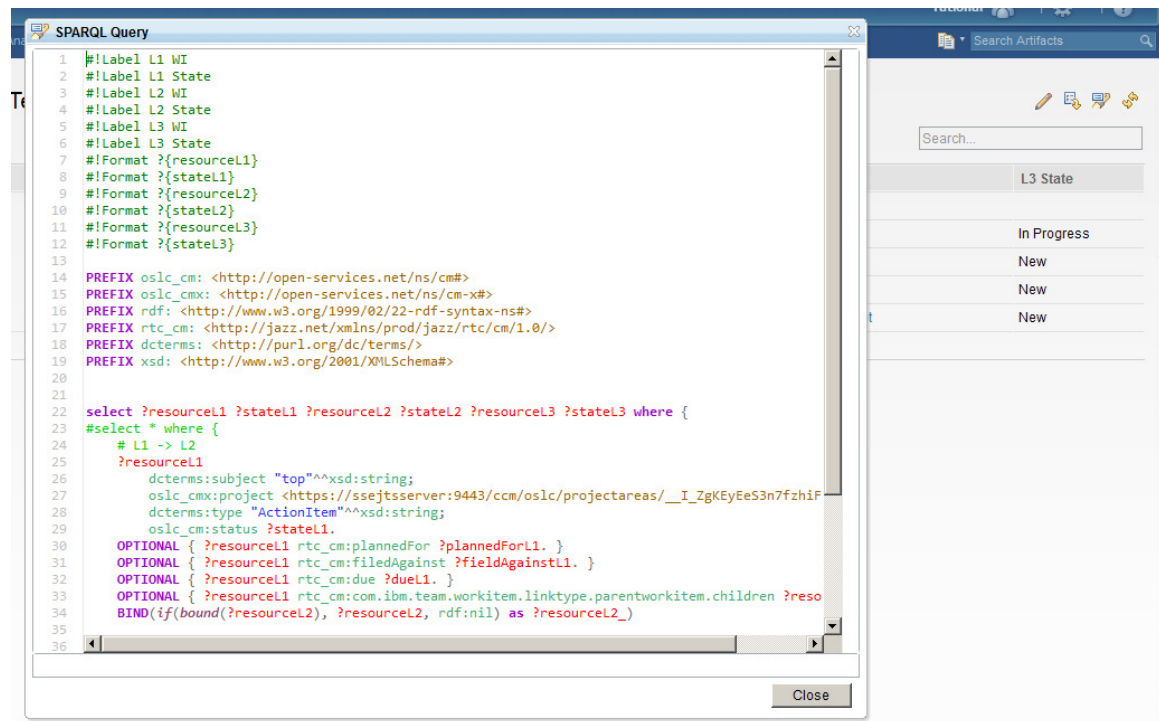
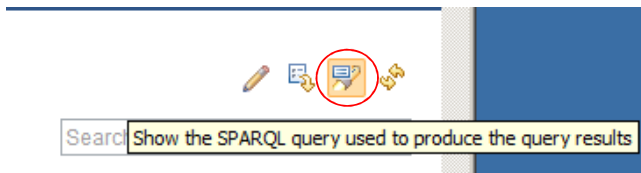
Search...

#	L1 WI	L1 State	L2 WI	L2 State	L3 WI	L3 State	L3 Change Sets
1	193: 2015 Model Development	New	194: Media Slot Development	New			
2			195: Monitor Development	In Progress	199: Monitor Plan	In Progress	
3					200: Monitor Main Development	New	
4			196: Audio Body Development	New	197: Audio Body Plan	New	Changes in: Car Audio Default Component - <No Comment> - rational - Jan 28, 2015 2:33 AM
5					198: Audio Body Main Development	New	
6	201: 2015 Development Plan	New					



## View Development Steps - Execute SPARQL (Cont.)

- RELM might alter SPARQL. ( for example, a parameter. ) The actual SPARQL to be executed can be verified from RELM Query result.



The screenshot shows a "SPARQL Query" window with a query editor and a results table. The query editor contains the following SPARQL query:

```
1 #!Label L1 WI
2 #!Label L1 State
3 #!Label L2 WI
4 #!Label L2 State
5 #!Label L3 WI
6 #!Label L3 State
7 #!Format ?{resourceL1}
8 #!Format ?{stateL1}
9 #!Format ?{resourceL2}
10 #!Format ?{stateL2}
11 #!Format ?{resourceL3}
12 #!Format ?{stateL3}
13
14 PREFIX oslc_cm: <http://open-services.net/ns/cm#>
15 PREFIX oslc_cmX: <http://open-services.net/ns/cm-x#>
16 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
17 PREFIX rtc_cm: <http://jazz.net/xmlns/prod/jazz/rtc/cm/1.0/>
18 PREFIX dcterms: <http://purl.org/dc/terms/>
19 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
20
21
22 select ?resourceL1 ?stateL1 ?resourceL2 ?stateL2 ?resourceL3 ?stateL3 where {
23 #select * where {
24 # L1 -> L2
25 ?resourceL1
26   dcterms:subject "top"^^xsd:string;
27   oslc_cmX:project <https://ssetjssserver:9443/ccm/oslc/projectareas/_I_ZgKEyEeS3n7fzhiF
28   dcterms:type "ActionItem"^^xsd:string;
29   oslc_cm:status ?stateL1.
30 OPTIONAL { ?resourceL1 rtc_cm:plannedFor ?plannedForL1. }
31 OPTIONAL { ?resourceL1 rtc_cm:filedAgainst ?fieldAgainstL1. }
32 OPTIONAL { ?resourceL1 rtc_cm:due ?dueL1. }
33 OPTIONAL { ?resourceL1 rtc_cm:com.ibm.team.workitem.linktype.parentworkitem.children ?reso
34 BIND(if(bound(?resourceL2), ?resourceL2, rdf:nil) as ?resourceL2_)
35
36
```

The results table on the right shows the following data:

Search...
L3 State
In Progress
New
New
New

## View Development Steps - Execute SPARQL (Cont.)

- Confirm the result from LQE Statistics
  - You can use LQE > Health Monitoring > Statistics > View As List > More Details
  - SPARQL to be actually executed is displayed with Execution Time. Adding a unique string in the first line might help you to see which SPARQL is executed from this page.

The screenshot shows the 'Statistics' page in the Lifecycle Query Engine. It features a navigation menu on the left with 'Statistics' selected. The main content area displays a small line graph and a 'View As List' link.



The screenshot shows the 'Completed Queries' table in the Lifecycle Query Engine. The table has columns for Status, User, Node, Start Time, Execution Time, Query, and Action. It displays four rows of successful query executions.

Status	User	Node	Start Time	Execution Time	Query	Action
Success	rational	nc9037034210	Jan 26, 2015 4:36:30 AM	0.005 seconds	SELECT ?subject ?predicate ?object WHERE...	More Details
Success	rational	nc9037034210	Jan 26, 2015 4:34:42 AM	0.004 seconds	PREFIX vvc: <http://jazz.ne...	More Details
Success	rational	nc9037034210	Jan 26, 2015 4:34:42 AM	0.004 seconds	PREFIX vvc: <http://jazz.ne...	More Details
Success	rational	nc9037034210	Jan 26, 2015 4:34:42 AM	0.004 seconds	PREFIX vvc: <http://jazz.ne...	More Details



The screenshot shows the 'Query Execution Details' page in the Lifecycle Query Engine. It displays a summary of the query execution, including status, user, node, start time, execution time, query type, and cache hit. Below the summary is the full query string with various prefixes.

Summary	Value
Status	Success
User	rational
Node	nc9037034210
Start Time	Jan 26, 2015 4:34:42 AM
Execution Time	0.004 seconds
Query Type	SPARQL
Cache Hit	False

**Query String**

```

PREFIX vvc: <http://jazz.net/ns/vvc#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX oslc_config: <http://open-services.net/ns/config#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX acp: <http://jazz.net/ns/acp#>
PREFIX ootb: <http://jazz.net/ns/vvc/ootb#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pd: <http://jazz.net/ns/pd#>
PREFIX trs: <http://jazz.net/ns/trs#>
PREFIX pd_ext: <http://jazz.net/ns/pd/extensions#>
PREFIX jazz: <http://jazz.net/xmlns/foundation/1.0/>
PREFIX jfs: <http://jazz.net/xmlns/prod/jazz/jfs/1.0/>
PREFIX ldp: <http://www.w3.org/ns/ldp#>
PREFIX pd_test: <http://jazz.net/ns/pd/extensions/test#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
    
```



## View Development Steps - Execute SPARQL (Cont.)

- LQE Query Result
  - URI replacement or column name replacement aren't done here. A raw query result from LQE is displayed.

Lifecycle Query Engine - 5.0.2 (LQE\_5.0.2-120141023-0815)

**Lifecycle Query Engine**

Home Health Monitoring Administration Query

**Query**

SPARQL Full Text Search

Enter a SPARQL query to query the data managed by Lifecycle Query Engine

```

PREFIX gslc_cm: <http://open-services.net/ns/cm#>
PREFIX gslc_cmx: <http://open-services.net/ns/cm-x#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rtc_cm: <http://jazz.net/xmins/prod/jazz/rtc/cm/1.0/>
PREFIX dcterms: <http://purl.org/dcterms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select ?resourceL1 ?stateL1 ?resourceL2 ?stateL2 ?resourceL3 ?stateL3 where {
#select * where {
# L1 -> L2
?resourceL1
dcterms:subject "top"^^xsd:string;
    
```

Run



Lifecycle Query Engine - 5.0.2 (LQE\_5.0.2-120141023-0815)

**Lifecycle Query Engine** rational | Log Out

Home Health Monitoring Administration Query

Query >

**Lifecycle Query Results** Execution Time: 0.005 seconds.

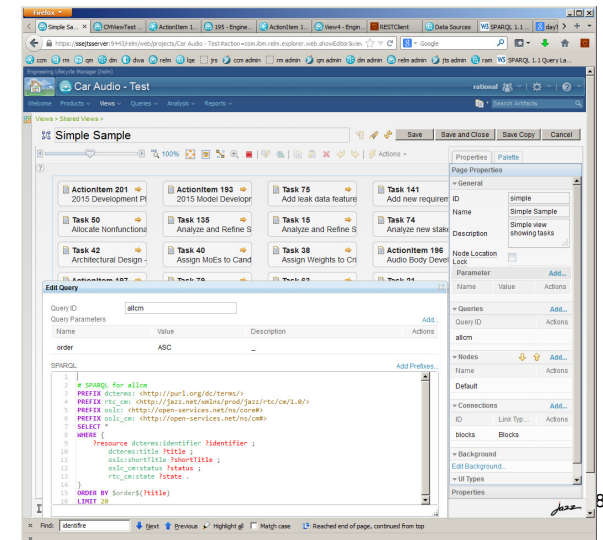
Previous | 1 - 6 of 6 | Next

resourceL1	stateL1	resourceL2	stateL2	resourceL3	stateL3
https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/193	New <http://www.w3.org/2001/XMLSchema#string>	https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/194	New <http://www.w3.org/2001/XMLSchema#string>		
https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/193	New <http://www.w3.org/2001/XMLSchema#string>	https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/195	In Progress <http://www.w3.org/2001/XMLSchema#string>	https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/199	In Progress <http://www.w3.org/2001/XMLSchema#string>
https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/193	New <http://www.w3.org/2001/XMLSchema#string>	https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/195	In Progress <http://www.w3.org/2001/XMLSchema#string>	https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/200	New <http://www.w3.org/2001/XMLSchema#string>
https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/193	New <http://www.w3.org/2001/XMLSchema#string>	https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/196	New <http://www.w3.org/2001/XMLSchema#string>	https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/197	New <http://www.w3.org/2001/XMLSchema#string>
https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/193	New <http://www.w3.org/2001/XMLSchema#string>	https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/196	New <http://www.w3.org/2001/XMLSchema#string>	https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/198	New <http://www.w3.org/2001/XMLSchema#string>
https://ssetjsserver.9443/ccm/resource/itemName/com.ibm.team.workitem.Workitem/201	New <http://www.w3.org/2001/XMLSchema#string>				

## View Development Steps - Fix Container

SPARQL

- Three objects are necessary to define View
  - Container : Layout to locate artifacts on View. (Grid / Tree ) SPARQL is bound to.
  - Node : Place holder to display artifacts. In order to show a hover window, SPARQL may be bound to.
  - UI Type : Definition to be displayed in Node. You often duplicate UI Type assigned to a default Node.
- If you need to design View quickly, it is suggested to use Grid as a Container and to bind SPARQL written in the previous step.
  - Using Sample view 's "Simple Sample" is an easy way.
  - You may need to write Connection(SPARQL) for Tree Container to find out child Nodes.



## View Development Steps - Modify Node/UI Type - Node

- Right click Node > Edit Node ... opens Node edit dialog
  - Using this is easier than using a right palette.
- You can duplicate it if you want to modify.
  - If being duplicated, you can recover Node to the initial state.
  - Once being duplicated, you need to change Node bound to Container ( by a property of Node in the palette.)
    - Example: Container's Node Type property. Note that Node might be fixed dynamically by using Node's Conditions and SPARQL.

### Containerのパレット

Parameter	
Node Type	<Not Specified>
Number ...	<Not Specified> Default
Show Grid	False

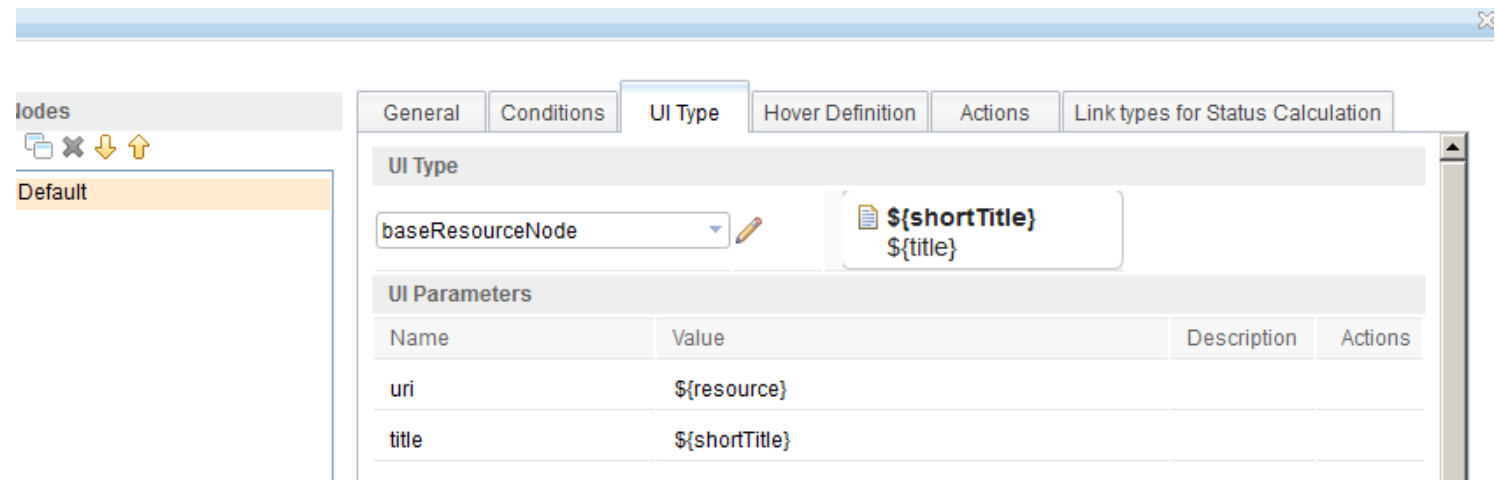
The screenshot illustrates the process of editing a node. On the left, a context menu is open for 'ActionItem 197', with 'Edit Node...' selected. A blue arrow points to the right, where the 'Nodes' dialog box is shown. The 'Nodes' dialog has a 'Duplicate' button highlighted in orange. The 'Nodes' list shows a 'Default' node with a 'Duplicate' button next to it. The 'Nodes' dialog has tabs for 'General', 'Conditions', 'UI Type', 'Hover Definition', 'Actions', and 'Link types for Status Calculation'. The 'General' tab is active, showing 'ID' as 'default' and 'Name' as 'Default'.

## View Development Steps - Modify Node/UI Type - Node (Cont.)

- General Tab
  - Specify ID or Name
- Conditions Tab
  - If data returned from SPARQL executed in Container satisfy the condition defined in this tab, this Node is used to display the returned resource. This might be used if the different visualization needs to be used in the same Container. ( In sample Views, Node for each artifacts, for example, Product, Requirement, Change Request, etc, are defined and the corresponding Node is used depending on rdf:type value.)

## View Development Steps - Modify Node/UI Type - Node (Cont.)

- Click UI Type tab shows UI Type currently bound to this Node. You can change UI Type here.
  - Its preview is available.
- You can overwrite the values of UI Parameters here.
  - You can pass SPARQL result value by using `${SPARQL variable name}`
  - Once UI Type design is change, you may need to change their parameters here too.
  - The parameters defined in UI type are displayed here.



The screenshot displays the configuration interface for a node's UI Type. On the left, a sidebar shows a tree view with 'Nodes' and 'Default' selected. The main area has several tabs: 'General', 'Conditions', 'UI Type' (selected), 'Hover Definition', 'Actions', and 'Link types for Status Calculation'. The 'UI Type' tab contains a dropdown menu set to 'baseResourceNode' and a preview window showing a document icon, the text `${shortTitle}`, and `${title}`. Below this is a 'UI Parameters' section with a table:

Name	Value	Description	Actions
uri	<code>\${resource}</code>		
title	<code>\${shortTitle}</code>		

## View Development Steps - Modify Node/UI Type - Node (Cont.)

SPARQL

- Hover window's content is defined in Hover Definition tab
  - The properties of Node or artifacts can be displayed. They can be selected from the list box.
  - You can use SPARQL to show some specific information. ( Until you will be familiar with SPARQL, it is suggested to use the predefined property values.)

The screenshot shows the 'Nodes' configuration window with the 'Hover Definition' tab selected. The window contains a table of predefined hover properties and an 'Add Hover Property' dialog box.

Label	Layout	Object or Query	Property Name
ID	Horizontal	Artifact Property	shortTitle
Title	Horizontal	Artifact Property	title
Description	Horizontal	Artifact Property	dcterms:description
Status	Horizontal	Artifact Property	oslc_cm:status
Related tasks	Vertical	Artifact Property	rtc_cm:com.ibm.team.workitem.linktype.relatedworkitem.related
Follow-on tasks	Vertical	Artifact Property	rtc_cm:com.ibm.team.workitem.linktype.blocksworkitem.blocks

Hide predefined hovers:  **Add Hover Property**

Label: \*

Layout:

Show a property value or a variable value from a custom query.

Artifact Property  Node Property  Custom Query

Property Name: \*

OK Cancel Cancel

## View Development Steps - Modify Node/UI Type - Node (Cont.)

- Define Action by a mouse click in Actions tab
  - Three actions, Open View, Reload Container and Reload Connections are available.
- Link types for Status Calculation
  - Get together Status from Workitems linked by a specified link types , and show the corresponding icon as a status.

## View Development Steps - Modify Node/UI Type - UI Type

- Click a pencil icon in UI Type tab opens its editor dialog
  - You can't modify UI Type of which name has "(System)"
  - Basically once UI Type is duplicated, you can modify it. In that time you need to change the assigned UI Type in the list box in Node dialog.

The screenshot illustrates the process of modifying a UI Type in the software. It is divided into three main sections:

- Left Panel (Nodes):** Shows a list of nodes with 'Default' selected. The 'UI Type' tab is active, displaying 'baseResourceNode' and a preview of a node with placeholders like `$(shortTitle)` and `$(title)`. An 'Edit UI Type' button is visible.
- Middle Panel (UI Types):** A list of available UI Types. A blue arrow points from the 'baseResourceNode' in the left panel to this list. A red circle highlights the 'Duplicate' icon next to the 'baseResourceNode (System)' entry. Below the list, a dropdown menu shows 'Duplicate baseResourceNode (System)', 'simpleResourceNode (System)', and 'requirementNode (System)'. The 'simpleResourceNode (System)' option is highlighted.
- Right Panel (Editor Dialog):** Shows the 'Class Type' dropdown set to 'com.ibm.reim.explorer.web.ui.view.nodes.ResourceNode'. Below it is a 'Preview' section showing a node with the same placeholders. At the bottom is an 'Attributes' table.

Name	Value	Description	Actions
uri	<code>\$(resource)</code>		
title	<code>\$(shortTitle)</code>		
summary	<code>\$(title)</code>		
width	160		
height	45		
overflow	clip		
fill	white		
stroke	#C5C5C5		
r	5		



## View Development Steps - Modify Node/UI Type - UI Type (Cont.)

- Define the values for the visualization in Attributes tab
  - The values you can change from Node. You can specify the content to be displayed and its attribute values, such as a colour.
  - Basically the necessary data can be obtained from SPARQL Query by specifying `${SPARQL variable}`

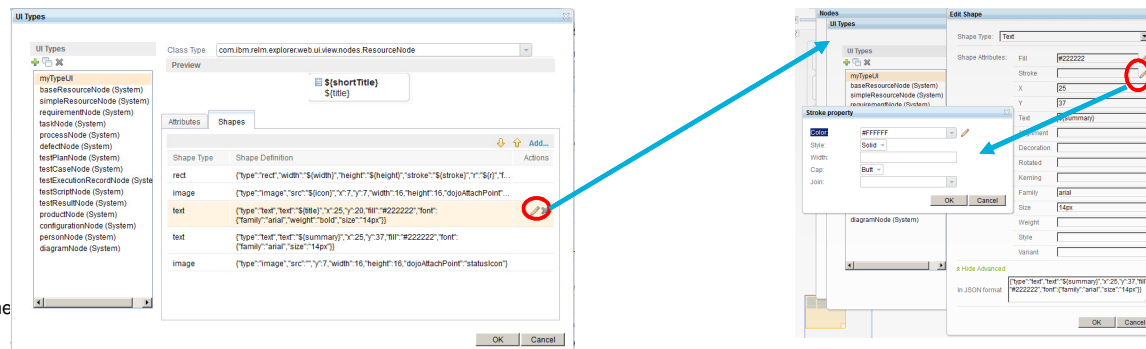
The screenshot shows the 'UI Types' configuration window. The 'Class Type' is set to 'com.ibm.reim.explorer.web.ui.view.nodes.ResourceNode'. The 'Preview' section shows a box with the text `${shortTitle}` and `${title}`. The 'Attributes' tab is active, showing a table with columns for Name, Value, Description, and Actions. The table lists attributes like uri, title, summary, width, height, overflow, fill, and stroke with their respective values and descriptions.

Name	Value	Description	Actions
uri	<code>\${resource}</code>	-	
title	<code>\${shortTitle}</code>	-	
summary	<code>\${title}</code>	-	
width	160	-	
height	45	-	
overflow	clip	-	
fill	white	-	
stroke	#C5C5C5	-	

## View Development Steps - Modify Node/UI Type - UI Type (Cont.)

- Design the content in Shapes tab

- Define the visualization object to be displayed here, such as Text, Eclipse, Circle, Image, etc.
  - If you want to show an image, you need to get its URI by using SPARQL anyway, and pass it to UI Type. So images should be placed in the place where it can be accessible by URI.
- Basically you need to specify x/y position, width/height size for the design.
- mltext(multi line text) can process a new line, but the new line feature should be enabled to mltext at the bottom in UI Type. ( because Shapes under mltext aren't automatically re-layout'ed even if mltext height will be grown.)
- The necessary data should be obtained from SPARQL through Attribute and Node.
- If you want to change the content to be displayed, you need to design UI Type to show the result of SPARQL through Node.
- You can specify various properties of each shapes. However, there might be no detailed information in Knowledge center.
  - If a pencil icon is available, you will click it to see the data input assist dialog.



## View Development Steps - Write Connection

- Define Connections to specify the lines between Nodes or Tree Container's child links
  - You can specify Connection for "Child Link" attribute in Tree Container. Also you can specify multiple Connections.
  - The value for Connection Title is displayed when the mouse is over the line.

Parameters for Query				Add...
Name	Value	Description	Actions	

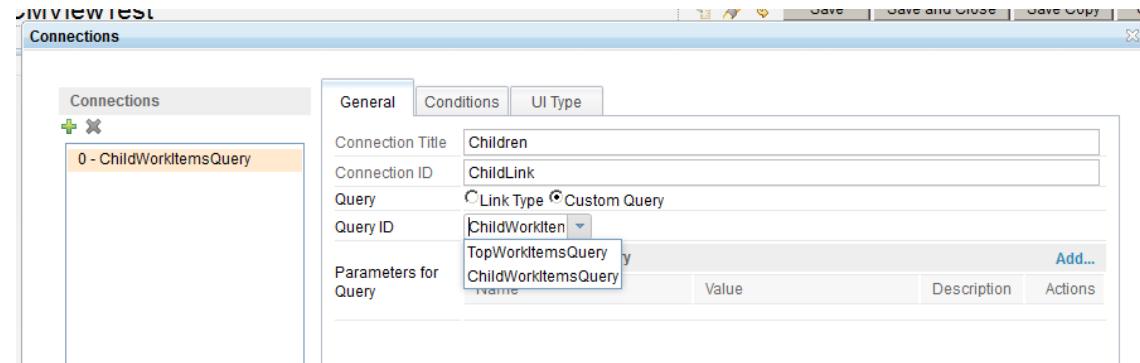
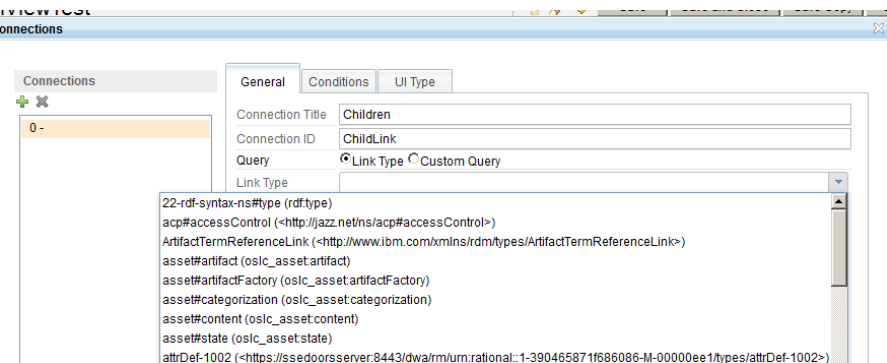
## View Development Steps - Write Connection (Cont.)

SPARQL

- If you want to see simple links, such as Workitem's parent / children, you don't have to write SPARQL. You can select it from Link Type list box.
- If you want to see not simple links but relation which satisfies some conditions within Nodes, you may use SPARQL to specify such condition.

Simple links

Conditions specified by SPARQL



## View Development Steps - Write Connection (Cont.)

SPARQL

- Sample SPARQL for Connection

- SPARQL which finds the child artifacts
- You may specify Connection by "children(rtc\_cm:com.ibm.team.workitem.linktype.parentworkitem.children)" in Link Type list box
- You need to write SPARQL for Connection if you need to show specific values obtained from SPARQL in Node(UI Type) or you need to use Node's Conditions. See below.
  - `$_com_ibm_relm_tree_values_$` is a code for the performance. (See Knowledge Center in detail.)

```
#ChildWorkItemsQuery
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
PREFIX rtc_cm: <http://jazz.net/xmlns/prod/jazz/rtc/cm/1.0/>
```

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

```
PREFIX osc_cm: <http://open-services.net/ns/cm-x#>
```

```
PREFIX dcterms: <http://purl.org/dc/terms/>
```

```
PREFIX osc: <http://open-services.net/ns/core#>
```

```
PREFIX osc_cm: <http://open-services.net/ns/cm#>
```

```
SELECT ?source ?target ?identifier ?type ?title ?shortTitle ?status ?statusColour
```

```
WHERE {
```

```
  $_com_ibm_relm_tree_values_$
```

```
  ?source
```

```
    rtc_cm:com.ibm.team.workitem.linktype.parentworkitem.children ?target.
```

```
  ?target
```

```
    dcterms:type ?type;
```

```
    dcterms:identifier ?identifier ;
```

```
      dcterms:title ?title;
```

```
      osc:shortTitle ?shortTitle ;
```

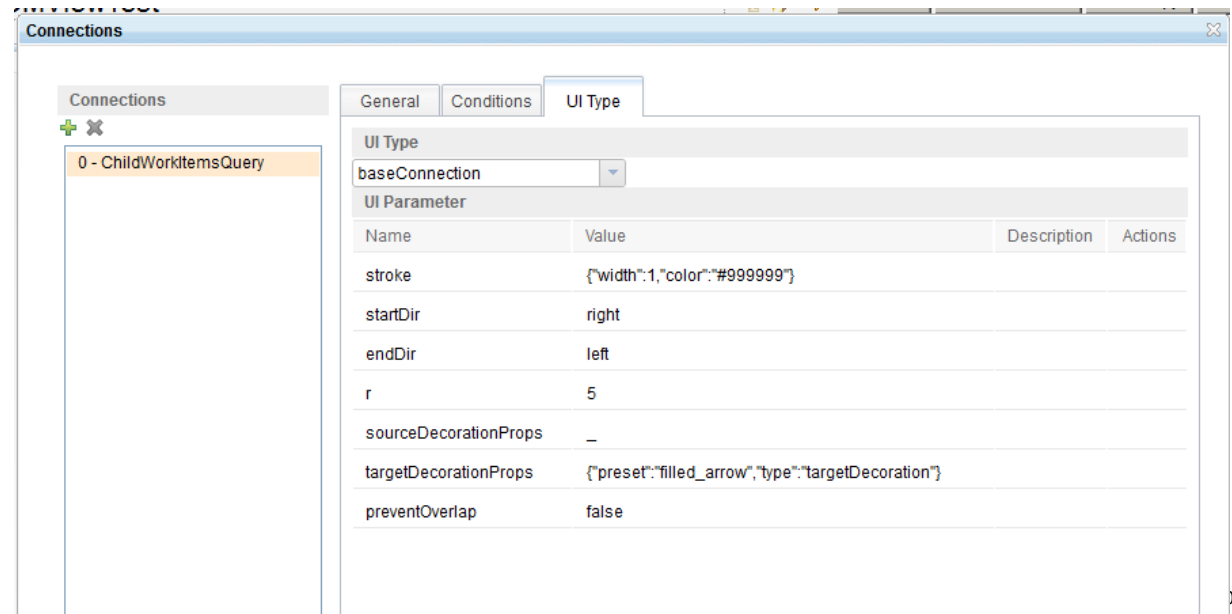
```
      osc_cm:status ?status.
```

```
  BIND(if(?status="New", "red", (if(?status="In Progress", "blue", "black"))) as ?statusColour)
```

```
}
```

## View Development Steps - Write Connection (Cont.)

- Specify conditions to be used in Connection in Conditions tab. (For example, you want to use the different color for Connection of V Process development and verification direction.)
- Specify the visualization attribute of Connection in UI Type tab. For example,
  - Type of line (straight or curly line, for example.)
  - Colour
  - Start of end position
  - Shape of an arrow
  - etc



## View Development Steps - Execute View

- Execute View to confirm the following
  - Necessary information ( characters, etc ) with the necessary attributes ( colour, size, etc ) are visualized.
  - Connection is displayed as being expected.
  - Contents in a hover window is what is supposed to be.
  - Action by click works as being expected.

The screenshot displays a software development environment. The main workspace shows a hierarchical diagram of development steps. A tooltip window is open over a 'Plan' element, showing the following details:

- Title:** ActionItem 201
- Summary:** 2015 Development Plan
- ChildLinks:** None
- Changes:** None
- Buttons:** Show Detail

The diagram in the background consists of several nodes:

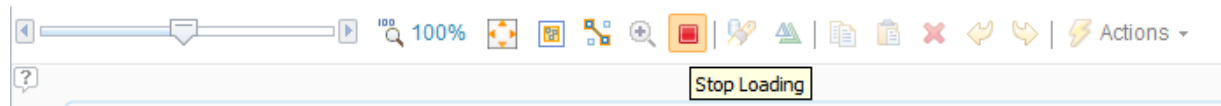
- 193:2015 Model Development (New)
- 194:Media Slot Development (New)
- 195:Monitor Development (In Progress)
- 199:Monitor Plan (In Progress)
- 200:Monitor Main Development (New)
- 196:Audio Body Development (New)
- 197:Audio Body Plan (New)
- 198:Audio Body Main Development (New)
- Change Sets (Action)

On the right side, the Properties palette is visible, showing the following configuration:

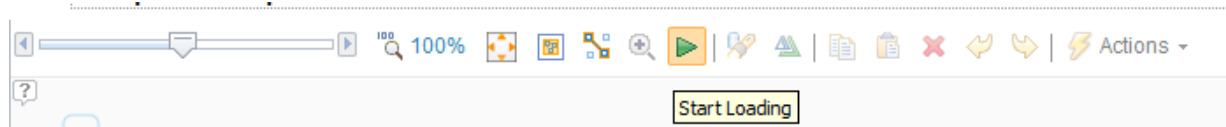
- Page Properties:**
  - General: ID (test1), Name (CMViewTest), Description (empty), Node Location Lock (checked).
  - Queries: TopWorkItemsQuery, ChildWorkItemsQuery, ChangeSetsQuery, ChangeSetsLinks4Hover.
  - Nodes: Name (empty), Actions (empty).

## View Development Steps - Execute View (Cont.)

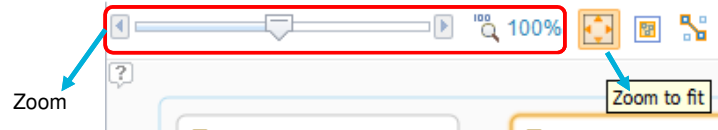
- If the display speed of View is slow due to the slow Query
  - The solution is to optimize Query to improve its performance.
  - You can use [View Development Steps - Execute SPARQL \(Cont.\)](#) 's " LQE > Health Monitoring > Statistics > View As List > More Details " to find which SPARQL is slow.
    - You will see the slow SPARQL or SPARQL to be called many times.
  - You can specify if SPRQL is executed or not when developing View ( this is a toggle button. )
    - Since Query isn't executed, Node/UI Type aren't rendered.



- Since Query is executed, everything is rendered.



- Zoom or Zoom to fit (changes a zoom rate to show whole view in the screen) can be used to limit the viewable area.





## Consider the usage of Jazz Reporting Service (JRS) ( v5.0.2 or later)

- Jazz Reporting Service (JRS)

- Supports search capabilities corresponding to RTC Search or RELM Query. As a data source, Data Warehouse and LQE can be used.
  - Data Warehouse - A target tool is only CLM. (You can use data or condition to be used in the existing dash board). Multiple project area is supported. Traceability within artifacts is supported.
  - LQE - Any tools which support OSLC/TRS in addition to CLM are supported. The data LQE indexes can be displayed. Flexible conditions can be specified. Multiple project area is supported. Traceability ( only OSLC link types) within artifacts is supported.
- Query for the corresponding data source is automatically generated and executed.
  - SQL or SPARQL is generated.
  - You can edit the generated command, but once you do it, you can't edit it by UI any more.
- You can download the output to Excel.

# Consider the usage of Jazz Reporting Service (JRS) ( v5.0.2 or later) (Cont.)

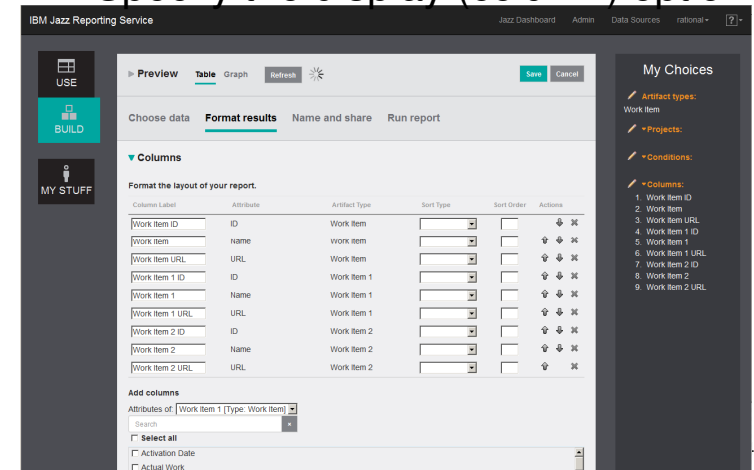
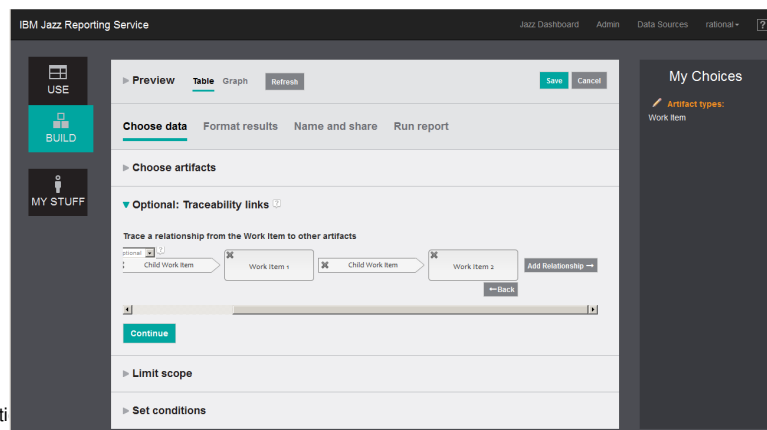
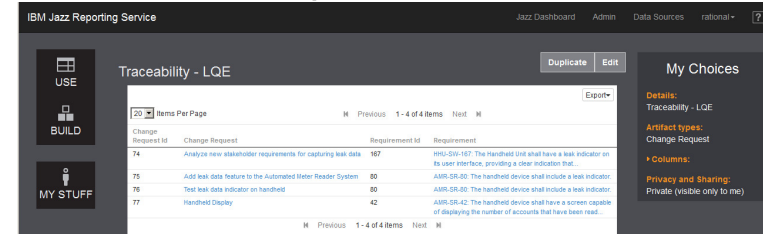
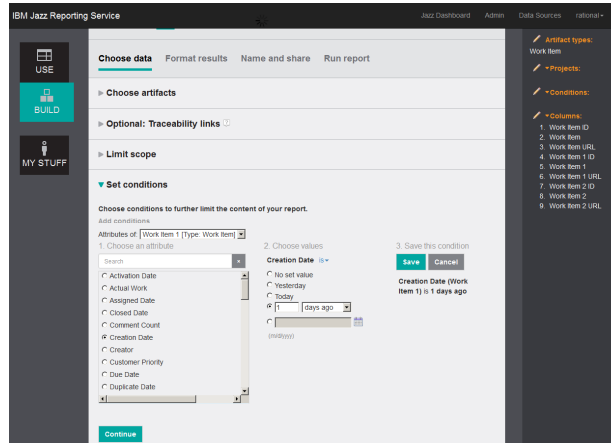
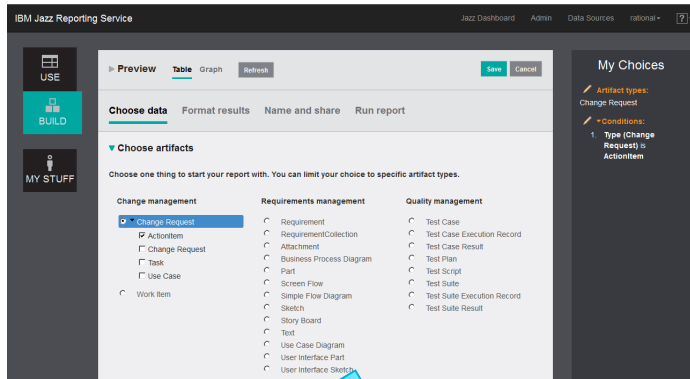
## 1. Specify the artifacts

## 3. Specify the condition

## 5. Execute a report

## 2. Specify the traceability

## 4. Specify the display (column) option





## Appendix: REST Client for Firefox

- Firefox add-on, RESTful API debugger
    - You can see a raw data by using this tool.
- Note: You need to specify the following headers.  
 OSLC-Core-Version : 2.0  
 Accept : application/rdf+xml



The screenshot shows the Firefox REST Client interface. At the top, the 'Request' section is active, displaying the method 'GET' and the URL 'https://ssetjsserver:9444/ram/oslc/assets/47994866-04B0-2D96-08FB-8C39D58DCDCE/1.0'. Below the URL bar, the 'Headers' section contains two entries: 'OSLC-Core-Version: 2.0' and 'Accept: application/rdf+xml'. The 'Body' section is currently empty. The 'Response' section is expanded, showing the raw XML response. The XML content includes namespace declarations for rdf, dcterm, xml:base, cam, cam:asset, cam:custom\_properties, cam:acc, cam:asset, and oslc, followed by a description of the resource and its content type.

This close-up view of the 'Headers' section shows two input fields. The first field contains the text 'OSLC-Core-Version: 2.0' and the second field contains 'Accept: application/rdf+xml'. Both fields are enclosed in light gray rounded rectangular boxes.

## Appendix:Developerworks : View development by RELM Artifact Element(AE)

- View can be developed by AE. ( This is a much easier way. )
- developerWorks articles ( All Japanese. )
  - <http://www.ibm.com/developerworks/jp/rational/library/relm/01/>
  - <http://www.ibm.com/developerworks/jp/rational/library/relm/02/>
  - <http://www.ibm.com/developerworks/jp/rational/library/relm/03/>
  - <http://www.ibm.com/developerworks/jp/rational/library/relm/04/>