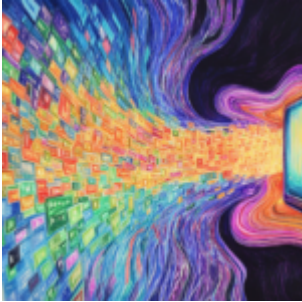


Decentralizing Metadata

written by DaltonRuer | January 22, 2025



Prerequisite

In my previous post on [The Importance of Metadata](#) I shared an example of some data values that looked crazy when viewed without the metadata about them. Then I shared how you could add that metadata into your Qlik Virtual Data warehouse files/tables. Normally I would simply put a blurb about “in my previous post ... blah blah blah.” In this you should consider it a prerequisite so you understand the data involved, the metadata involved and more importantly: the concept of decentralizing the metadata to other developers not just yourself. If you don’t understand those things this post might go way over your head.

One person that totally understood was [Armando Lopez](#). Before there were even any other comments on my post, Armando wanted more. “Qlik Dork it’s awesome that you can decentralize all of that delicious metadata with other developers, those who can read script files, and those who can understand the Data Model Viewer ... but what about my users? They are BEGGING ME FOR IT?” Ok, I shared the image so you know that’s not exactly how he phrased it. I might have taken liberties for a smile.



Armando Lopez • 2nd
Director of Data and IT

3h ...

Is there a way to display this to users? I've gotten asked "how is this field calculated" or "where does this data come from". It would be cool to display this metadata for users too.

Like · 1 | Reply · 1 Reply

Armando asks about decentralizing metadata to end users as well as developers

Decentralizing Metadata to End Users

I've never met Armando's end users. I've never even met Armando. But with my ESP I can tell you exactly how the conversations went with Armando, and his end users, and between your end users and you -- "Our company has many different ways that the same things can be measured. This report says the KPI is 17, that report says the KPI is 20 and that report says the KPI is 14. I can't take ACTION because I don't TRUST what I'm looking at." Or worse, every meeting ends in a fight over the numbers and the only action items are "Dalton you go off and get the detail to prove to me why the number is 17, and Armando your action item is to prove why the number is 14."

I was in that environment for 30 years ... so, while I have never met Armando's end users I do know their pain. There is no such thing as a "single source of truth." There is a single source of truth by department, or organization, or even managers. Teams/people call them the same thing, but you absolutely must know which "version of the truth" you are seeing on the screen. If you have no idea, then you can't trust it, and if you don't trust it you certainly can't act on it.

Please do not miss this point

In most environments the problem is even worse ... I go back and run my SQL query and I get 18, not 17 anymore. Armando runs his SQL query again and he now sees 16 instead of 14. One of the most important aspects of using Qlik Virtual Data warehouse tables/files (QVDs) in the first place ... is the fact that the data in them is a point in time snapshot and is CONSISTENT. While our calculations are obviously different, the data itself is CONSISTENT. Me and Armando can meet, share some cookies, enjoy a Mountain Dew and discuss our EXPRESSIONS/CALCULATIONS. Without having to pull our hair out, or doubt ourselves. Because each of us can absolutely reproduce the same numbers, without worrying that the source system values are constantly in flux.

Please do not miss this point either

As a customer who loved QlikView, I saw Qlik Sense for the first time in November 2014 and it changed my world. Not because it was new. Not because of responsive visuals that resized and changed what was displayed as the user changed what platform or size of screen they were using. The reason was simple ... MASTER ITEMS. The ability to add the metadata and the heavy duty coding into a simple consumable Dimension or Measure that application designers could easily use to drag/drop and create screens and because END USERS could easily hover of a field and SEE THAT METADATA.

As a QlikView customer you better believe that I had lots of "super users." They could copy objects and add a column. "Ooooh." "Ahhhh." They couldn't change the KPI's because they couldn't recreate different versions of expressions they weren't coders. Unless I hand coded metadata into each object, nobody was able to know for user which version of truth they saw. Master Items changed all of that. Now designers/super users can drag/drop them and see the definitions to know "oh I

see Dalton was using this way that Division 1 considers to be right, and my department is in Division 2 and considers this way to be right and that's what Armando was showing.

Getting personal for a moment ... Master Items were so important to me, that I joined Qlik just months later to help customers globally understand their game changing power. So, you better believe I jumped in joy when Armando proactively asked for what I already had planned: "A way to expose that metadata to end users." ☐

Example

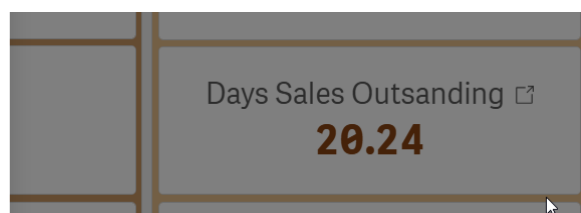
In the SAP world there are literally dozens of ways to calculate a value called Days Sales Outstanding and it is one of the biggest KPI's organizations track. You can imagine how many fights might ensue in a meeting if nobody understands which of the dozens is being use. But what a difference it makes being able to hover over a value and see that in Dalton's KPI it is a "weighted" calculation, where Armando's is a "raw" value with no regard for the transaction value.

Measure



Days Sales Outstanding

This is a WEIGHTED DSO calculation. The value is the sum of DSO # Days to Clear * the Line Amount and then the divided by the sum of the Line Amount



Weighted DSO Calculation with metadata visible thanks to Master Item

Dimension



DSO Range

This is the actual number of days payment was outstanding and has no regard for the amount.

A different DSO calculation that is NOT weighted and the metadata is clear

QVD to Master Item

Now that we are all on the same page:

□We believe that decentralizing metadata is important

□We know how to do that in the QVD files themselves

□We know that Qlik Sense provides the means to share that information to designers and end users

The next obvious thing we need to do is move all that wonderful metadata about the fields into some Master Items. I could have responded to Armando “Just copy/paste it one field at a time.” It’s not like we provide any means of Automating that task.

Oooh wait a second, we do have that Qlik Application Automation thing-a-ma-jiggy. Let’s think this through for a second because we just might be on to something here.

The Task of Decentralizing the Metadata

The manual task to decentralize the data even further to end users would simply be to go through all of the fields in an application one at time, copy the metadata, create a new Master Dimension, then paste the “comments” into the Description field, and then copy the tags for the field and paste them into the “tags” for the new Master Dimension. Oh and press Create of course.

Create new dimensions

Single Drill-down Cyclic

Filter by table

All tables

Q Search

Age

Arrival

Discharge

EncounterID

Name

PatientID

Field

fx

Name

Label expression

fx

Description

fx

Dimension color



Tags

+

Glossary terms

+

Create

Close

Blank screen to Create a New Master Dimension

Some of you reading are thinking “that doesn’t really seem so bad, I can do that manually with no problems.” At least I don’t have to think about using that Qlik Application Automation stuff. I’ve never really understood it, and who knows, it might make my head hurt. ☐

Others of you may build applications that might have hundreds of fields in them. Like the **SAP Accelerator for Orders to Cash** application that I built to help our customers accelerate their journey. That might really stink to have to go through that manual process 1 field at a time. So, maybe, just maybe, this is a good time to finally take a look into Qlik Application Automation.

Quick Summary			
Table Name	Q	# Fields	# Rows
Totals		936	3,818,592
Document Terms		12	1,589,497
Terms		5	1,589,497
ORG		4	263,310
Document Metadata		4	175,000
Billing Documents		94	52,254
Schedule Lines		41	29,952
Sales Items		129	26,332
Document Status		11	26,331
Document Text		8	25,000
Open Orders		16	24,327
OFIAR4		82	14,208
Materials		160	1,166

Quick Summary screen that utilizes system tables/fields to help the end user understand the volume of data inside the application.

Automating the Task to Decentralize the Metadata

In this post I am not going to teach you everything about Qlik Application Automation. [I've already created an entire playlist on YouTube that you can watch to learn the basics and the concepts that you can watch.](#)

What I will do is walk you through the algorithm/logic I used to automate the process of decentralizing the metadata with end users. As I'm going to share this Qlik Application Automation with you I wanted to start by displaying a picklist where you could simply choose the application you wanted to do this for. I could have been cheesy and hardcoded a single

application, but you know that's not how I roll.

Application: *

(QCDI first Financials) [97cb64df-6437-4c2d-a96c-75a519cf2a1f]

Pharma Sales [8d61e440-839e-41f0-bd95-6a503b34a8ad]

Population Health SDoH Geo [d12f842d-a91f-4425-9391-d8395deb73b1]

Predict HQ [69a142dc-dbe9-4107-8327-d64da1a22af4]

Previous_and_Peek [628170ca-ab99-4559-90ee-b18b79a2b737]

Product Sales_Enablement_Sept_2020_ [2929f264-3386-410f-85bb-c18693a4b718]

QVD Extractor (SaaS) [7b2c4acd-43b9-40c0-b91e-1d0bd5dd5804]

Qlik Answers [a4a97aa0-e91e-46f3-8c73-a17f262a2112]

Qlik Cloud Ready Demo App [e22dbc85-dc91-4c27-9072-c4fcf2fbde37]

QlikRest [40487e5e-97ad-4379-90aa-ea7841a2938f]

Qlik_Community_Tester_For_Questions [de637c34-6499-4482-b5ae-08c202496370]

Rank [c5090387-e6f3-4fe1-bea5-2f94f8b374aa]

Read from QVD [1badf04f-0d8a-4a1a-a4e8-9cedaec8d8ba]

Ready for SaaS Demo [cfad9175-d084-49f4-b5be-2381eb0f0ff0]

Retail Sales Analysis - Copy For Export [a58a4963-71b4-4c02-80a4-f4b142a80655]

SAP Financials [bec7334e-59d9-4d55-a6b0-1404254b20e0]

SAP GL Details [565f5052-42ed-4114-b146-7ceeab5d7458]

SAP Inventory Management [5d9c1316-585f-43b3-aca3-d91b4221faa2]

SAP O2C (AWS Redshift Demo) [842f1790-d439-4b71-959f-50c066d7c947]

SAP O2C -AZure (Dalton) [67408799-dd77-4fc4-ad0a-8bcaa8cdf67c]

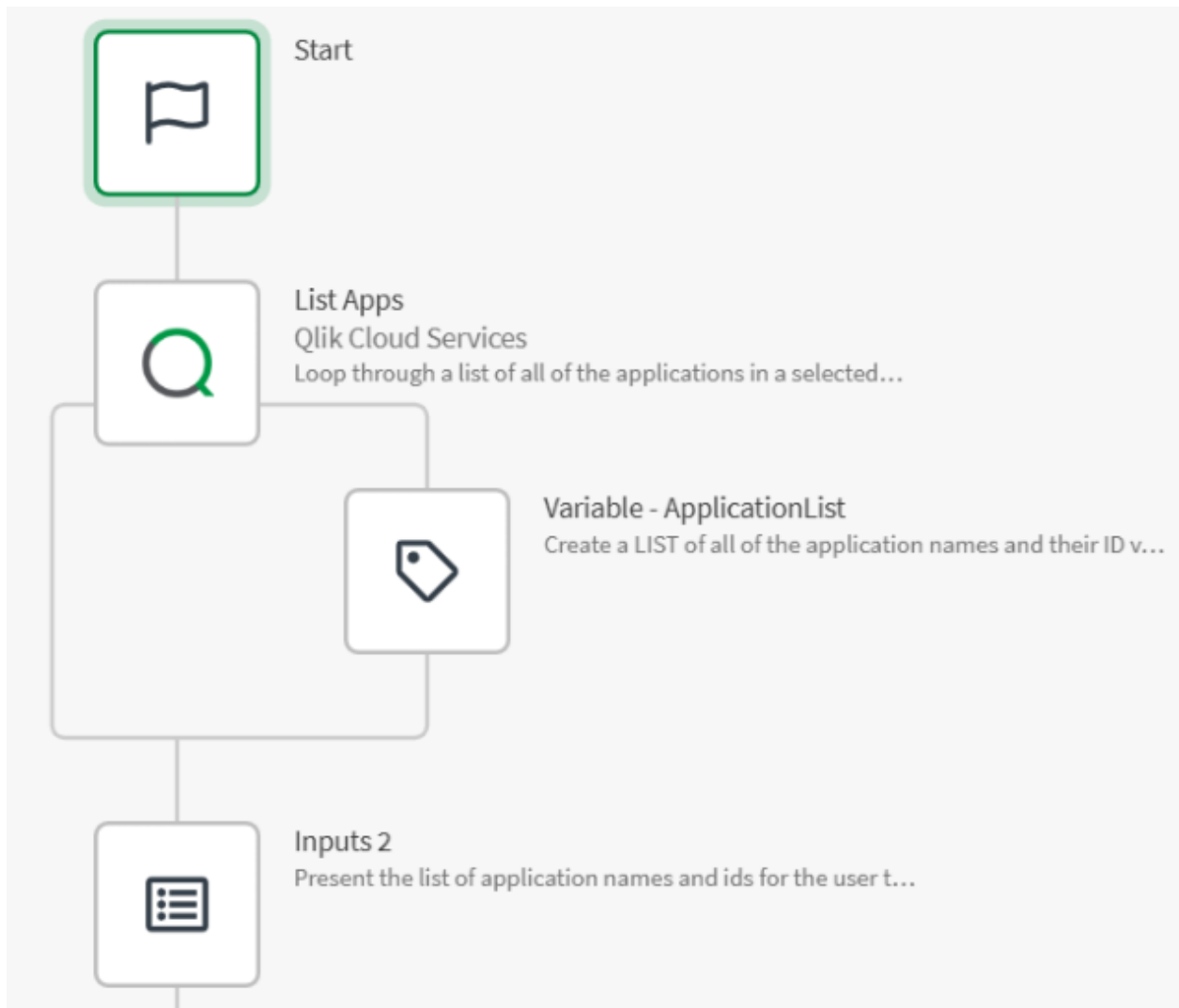
SAP O2C -Azure OpenAI [81f5d266-daa4-451c-974b-810df28b988b]

Dropdown list showing all of the application names and ID's for a particular space

Step 1

How did I present the list? Easy breezy lemon squeezy. I use a "List Applications" block that iterates through all of the applications. In my case the list of "all applications" is a bit overwhelming. So, I used the setting in that "block" to just choose my "personal" space where I knew the applications

were that I would test with. As it looped through each one I simply added the Application Name and the ID to a Variable that itself was a list. Notice in the image above that I wrapped the ID in square brackets “[]” that’s important momentarily.



Screenshot from the Qlik Application Automation that lists the applications and builds the list previously visualized

Step 2

Next I simply created an Input Block. Normally when you see them, like you did if you watched my playlist for Qlik Application Automations, you simply see an input field. In our case, I needed to utilize the Dropdown type instead. As for

the options ... too easy. I just handed it the list of application names and ids' created above. Voila easy input box that I (you) can click and choose the desired application.

The screenshot shows a configuration panel for an input field. At the top, there is a tabbed interface with 'Inputs' selected, 'Settings', and 'History'. The main configuration area is titled 'Application: *' and includes a trash icon. It contains several sections: 'Label *' with a dropdown set to 'Application:'; 'Help text' with an empty dropdown; 'Input type *' with a dropdown set to 'Dropdown'; and 'Options' with a dropdown set to 'implode → Application List'. Below the options dropdown, there is a text instruction: 'Enter a comma separated list of values (e.g. option 1, option 2, option 3) or a list of objects where each object has keys "id" and "name". Only the first 1000 options will be displayed.' At the bottom, there is a checked checkbox labeled 'This field is required'.

Screenshot from the Qlik Application Automation that uses the list for the dropdown input

Step 3

Now that an application was chosen, I need to create a variable that would contain the Application ID. That's what other needed blocks need as input. I previously made a point of indicating that I was wrapping the Application inside of square brackets. That was so that I could easily use a REGEX

expression to parse the ID out from inside those brackets.

The screenshot shows a configuration interface for a variable named "ApplicationID". It has two tabs: "Inputs" (selected) and "History". Under "Inputs", there is a text field containing the value "AB0DF5B2-4917-43BD-BDF5-60D2E2520B57" and a "Manage variables" button. Below this, under "Operations on ApplicationID", there is a dropdown menu set to "Set value of ApplicationID". Underneath, there is a "Value" field containing the regex expression: "{regexparse: { \$.inputs2.'Application:' }, '\\((.*?)\\}'.

Screenshot from the Qlik Application Automation that uses a regex expression to parse out the selected application id from the dropdown list

Don't be too impressed. I knew I needed a REGEX expression. I knew it would work. But I don't recite REGEX syntax in my sleep. So, how did I come up with the right set of jibberish?

Chat GPT of course. ☐

regex expression to pull all text inside of []



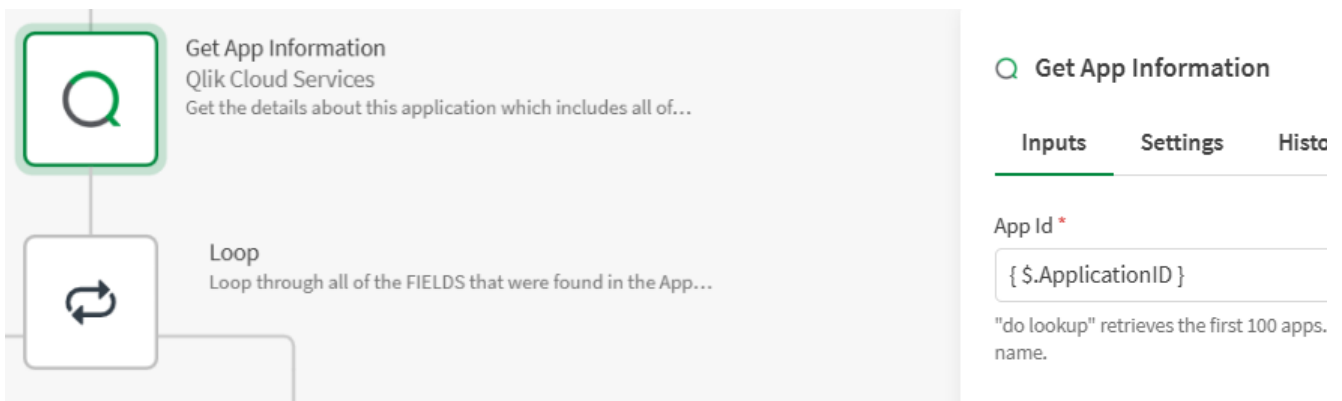
To extract all text inside square brackets [] using a regular expression, you can use:

```
regex Copy Edit  
  
\\((.*?)\\
```

Screenshot from Chat GPT showing one of the things it is really good at ... building code you may not be strong in

Step 4

The next step was to have the Automation “open” the selected application and get the important information in it ... LIKE THE LIST OF FIELDS. Then I simply looped through them.



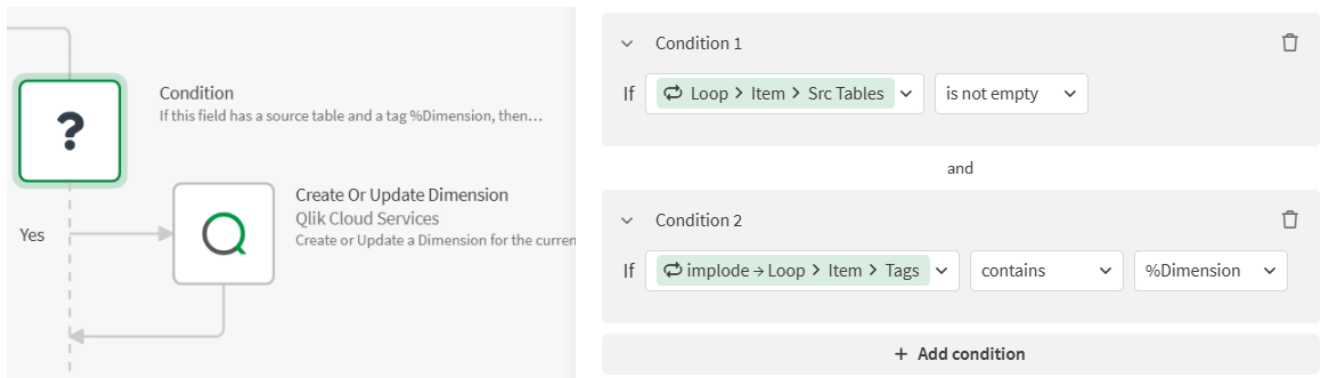
Screenshot from the Qlik Application Automation showing you can ask for the application information about your selected application and then loop the fields it contains

Step 5

Before you say “I get it ... then you simply used a block to create a Master Dimension for every field” ... there are 2 things you should consider:

1. Qlik maintains a series of “system tables” and “system fields.” They are values you can reference inside of your application. For instance that Quick Summary table object I used to illustrate that my SAP Accelerator had 963 fields. I didn’t really want those to become Master Dimensions.
2. That value 963 fields might be a little overwhelming for application designers or end users to scroll through.

With both of those things in my mind I decided that I only wanted to “Create or Update” the Master Dimensions for those fields that had a value for the table they were sourced in, and that had a tag value of “%Dimension” that I had to use inside my load (creation) script as a clue ... “Hey Automation this field is important so please create a Master Dimension” for it.

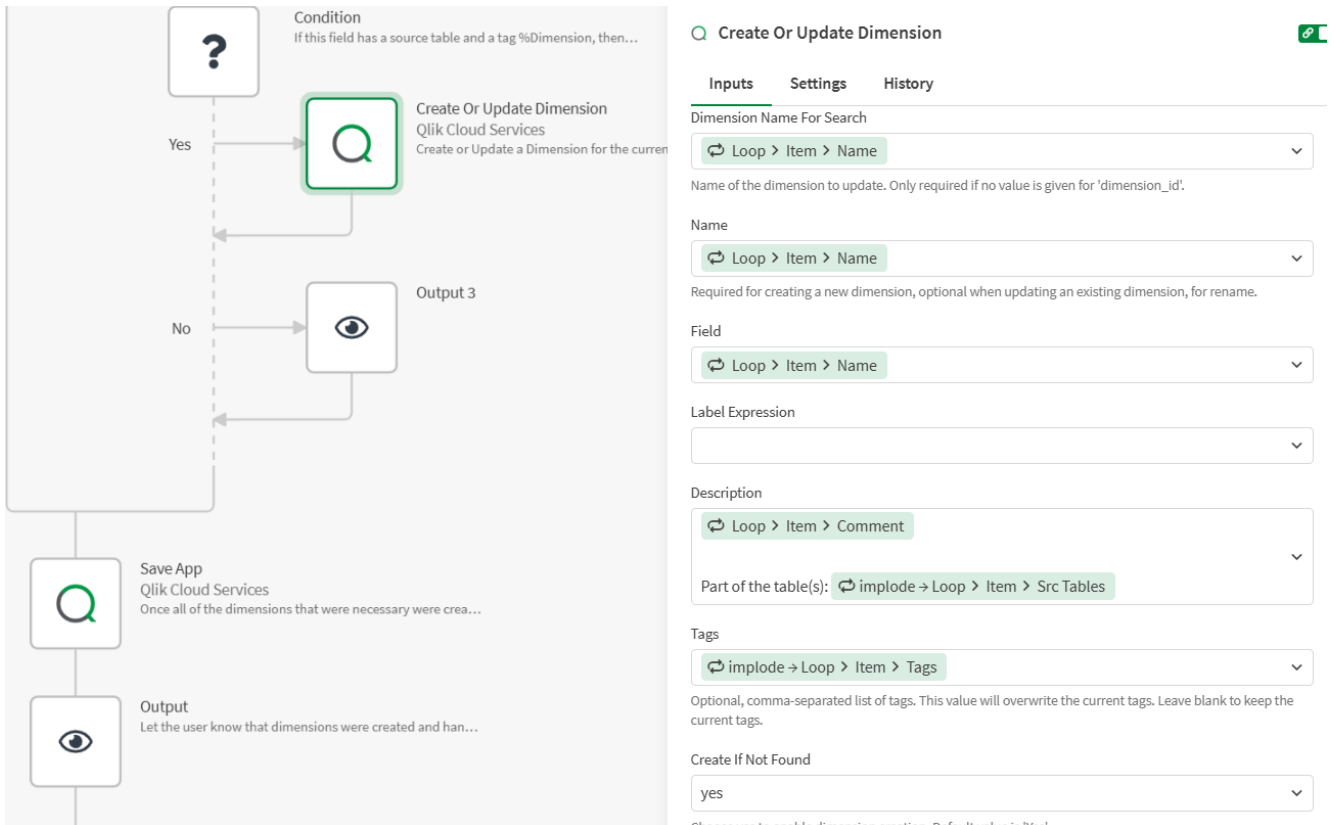


Screenshot from the Qlik Application Automation showing you can apply conditions so you aren't just creating dimensions for every field in the application

Step 6

Immediately upon reading my conditions you might have thought of your own. That's a great thing. The great thing is that you easily add as many conditions and decide if they are AND's or OR's. I simply chose something simple to show you the logic.

The most important part of this Qlik Application Automation is this step ... Creating or Updating a Master Dimension with all of that previous metadata. In the image you can see that I simply said to create the Dimension with the same name as the field, decentralize the metadata in the Description field, and then add the tag metadata in the tags field. If the Master Dimension was already created, it is updated. If the Master Dimension did not exist, it will be created.



Screenshot from the Qlik Application Automation showing the good stuff ... creating a master dimension to further decentralize the metadata for end users and designers

Hands on to Decentralize your Metadata

I built this Qlik Application Automation to be shared. I wanted y'all to actually to be able to use it. So, here is the link where you can pull this down from a GitHub repository I created to share Qlik Application Automations.

[Click here to go to the Qlik Dork's GitHub Repository to get the Create Master Dimensions automation](#)

If you really are new to Qlik Application Automations you might be saying "Gee, thanks for a JSON file Qlik Dork now what do I with it?" Well be sure to read this [Qlik Community post that walks through how to import the JSON you download from me, into your tenant](#). Also be sure and watch the video

playlist I shared earlier in this post to understand more about them.

Summary

The purpose of this post was to help you further value and appreciate the metadata you have, and help you further decentralize the metadata to your users as well as developers. If you are trying to increase adoption, and grow trust to encourage action ... this is something you really need to focus on. I used a Qlik Application Automation to greatly simplify and speed up the initial process. When a new application is built from the QVD's, simply load them, and before you start building screens ... run this automation to decentralize the metadata into Master Dimensions to help you get started.

The End

Not really. Because I know some wise guy out there is going to respond in comments here, or on LinkedIn and say "Well that was really slick. But I've got a manager who really wants all of this deliciously powerful metadata available outside of the Qlik environment." I hope they don't because my goodness I'm not sure how I would respond. ☐