Importing DAD Data

DAD has the ability to import data in bulk from external sources.

Process

There are two stages to the import – Putting the component data in a flat list into a table on the DAD database, and then importing the data

1 – Importing the data to a flat table

The main table is called ***tblAPI_Import***. It has the following columns which are strings unless specified.

External_id (integer): (*Optional*). This value is not stored in DAD, but it just copied for the relevant rows in the tblAPI_Processed and tblAPI_Results tables for easy external matching.

Tag_Name: The Tag Name in DAD. If the matching component has a different tag Name in DAD, then this will be updated to match this value. New components will use this as the tag.

DAD_Portal: This is not used. It is just to the import table format is the same for v12 and v13 imports.

DAD_Layer: The DAD Layer the component is in.

MatchingAttribute_Name: There are 4 different options this could have:

1) The attribute in the Type tree to match for the component. E.g. `Note`
2) `DAD:Tag Name` – This will match on tag name
3) `DAD:id` - This will match on the internal DAD id. This can be retrieved via the DAD spreadsheet
4) `DAD:Always Create` – This will not try to match to an existing DAD component – it will always be created using Tag_Name as the description.

MatchingAttribute_Value: The value to match on. For the 4 cases above the expected value will be

1) The value stored in the DAD attribute
2) The tag name in DAD
3) The internal Dad id (number)
4) Not used since not matching so just use the empty string.

LocationParentAttribute_Name: (*Optional)*. The name of a Location attribute to match for a parent folder to either put newly created components in, or to move existing matched components to. If nothing specified then newly created component added to a folder called "Import"

LocationParentAttribute_Value: (*Optional*) The value stored in DAD in the LocationParentAttribute_Name attribute to match on.

TypeParentAttribute_Name: (*Optional)*. The name of a Type attribute to match for a parent folder to either put newly created components in, or to move existing matched components to. If nothing specified then newly created component added to a folder called “Import”

TypeParentAttribute_Value: (*Optional*) The value stored in DAD in the TypeParentAttribute” attribute to match on.

Attribute1_Name: There are 2 different options for this

1) The name of the first attribute in the Type tree which will be updated.
2) `id:XXXX` – Where XXXX is the internal DAD id for the type tree attribute.

Attribute1_Value: The value for the first attribute which will be updated

…

Attribute10_Name, **Attribute10_Value**

Attribute1 – 10 are all optional. They don’t need to be filled in.

2 – Running the Import

The import is run by executing the stored procedure

Exec spAPI_Import

This goes through the tblAPI_Import table and applies the following

- If there is 1 matching component, then it updates the Tag Name (f Different to Tag_Name) and the attribute values if different
- If there if >1 matching component, then is does nothing for this line
- If there is 0 matching components it will make the component in the folders (Default “Import”) in both Type and Location tree

3 – Check Results

The results for each line is stored in a table called

tblAPI_Import_Results

This lists the output for each line with any problem listed. Errors include

- Error: Invalid Layer Name
- Error: The Matching Attribute (if not the 3 “DAD:xxx”) does not exist in the Type tree
- Error: More than 1 matching components found for a line
- Info: Component Created, Tag Name updated, Attribute Value updates, Location or Type parent changed.

Examples

Creating New Components

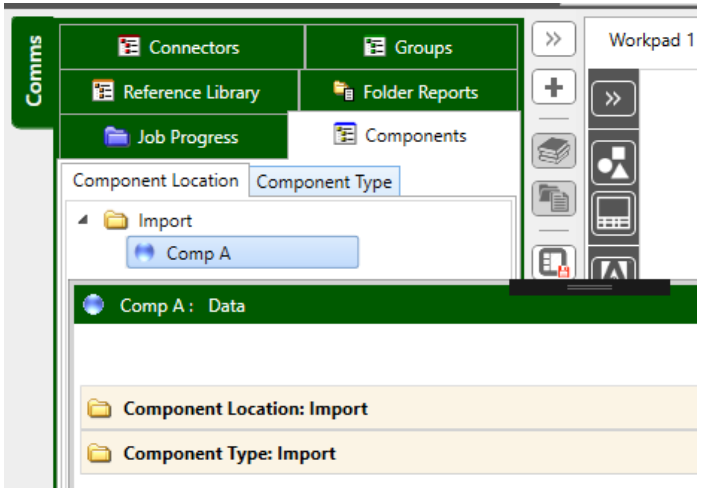
New Component with no parent or attribute information

This example shows 1 component. Since it doesn't match any existing component, it will be created. Since no Location\Type Data specified, it will be created in "Import" directories.

Header Data					
External_id	TAG_Name	DAD_Portal	DAD_Layer	MatchingAttribute_Name	MatchingAttribute_Value
122	Comp A	SIM	Comms	DAD:Tag Name	Comp A

```
Insert into tblAPI_Import(External_id, TAG_Name, DAD_Portal, DAD_Layer, MatchingAttribute_Name, MatchingAttribute_Value)
Values ('122','Comp A','SIM','Comms','DAD:Tag Name','Comp A')
```

This gives this component



And the results table shows the results that is was created

```
select * from tblAPI_Import_Results where External_id = 122
```

DAD_UID	External_id	DAD_id	Date	Comment
4	122	145045	2019-06-25 09:58:00	Created

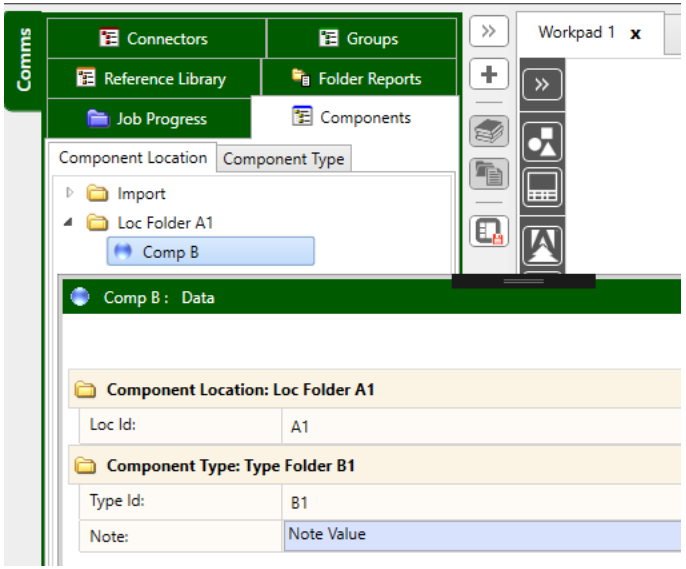
New Component with parent and attribute information

For this, in DAD, a Location attribute called “Loc id” was created and added to a location folder with a value A1. “Type id” was created and given to a type folder with value B1. A Type attribute “Note” is also in DAD. The import data was :

Header Data						Location/Type Data				Attribute Data	
External_id	TAG_Name	DAD_Portal	DAD_Layer	MatchingAttribute_Name	MatchingAttribute_Value	LocationParentAttribute_Name	LocationParentAttribute_Value	TypeParentAttribute_Name	TypeParentAttribute_Value	Attribute1_Name	Attribute1_Value
123	Comp B	SIM	Comms	DAD:Tag Name	Comp B	Loc Id	A1	Type Id	B1	Note	Note Value

```
Insert into tblAPI_Import(External_id,TAG_Name,DAD_Portal,DAD_Layer,MatchingAttribute_Name,MatchingAttribute_Value,LocationParentAttribute_Name,LocationParentAttribute_Value,TypeParentAttribute_Name,TypeParentAttribute_Value,Attribute1_Name,Attribute1_Value) Values('123','Comp B','SIM','Comms','DAD:Tag Name','Comp B','Loc Id','A1','Type Id','B1','Note','Note Value')
```

This gives this component



So it matched the two parent folders and used them as parents. The Note value was then added to the component with the passed value.

The results table shows that it matched the folders and updated note

```
select * from tblAPI_Import_Results where External_id = 123
```

DAD_UID	External_id	DAD_id	Date	Comment
6	123	0	2019-06-25 10:42:00	Matched folder "Loc Folder A1" with Location Parent Attribute Name - "Loc Id" (A1)
6	123	0	2019-06-25 10:42:00	Matched folder "Type Folder B1" with Type Parent Attribute Name - "Type Id" (B1)
6	123	145053	2019-06-25 10:42:00	Created
6	123	145053	2019-06-25 10:42:00	Attribute "Note" added with value "Note Value".

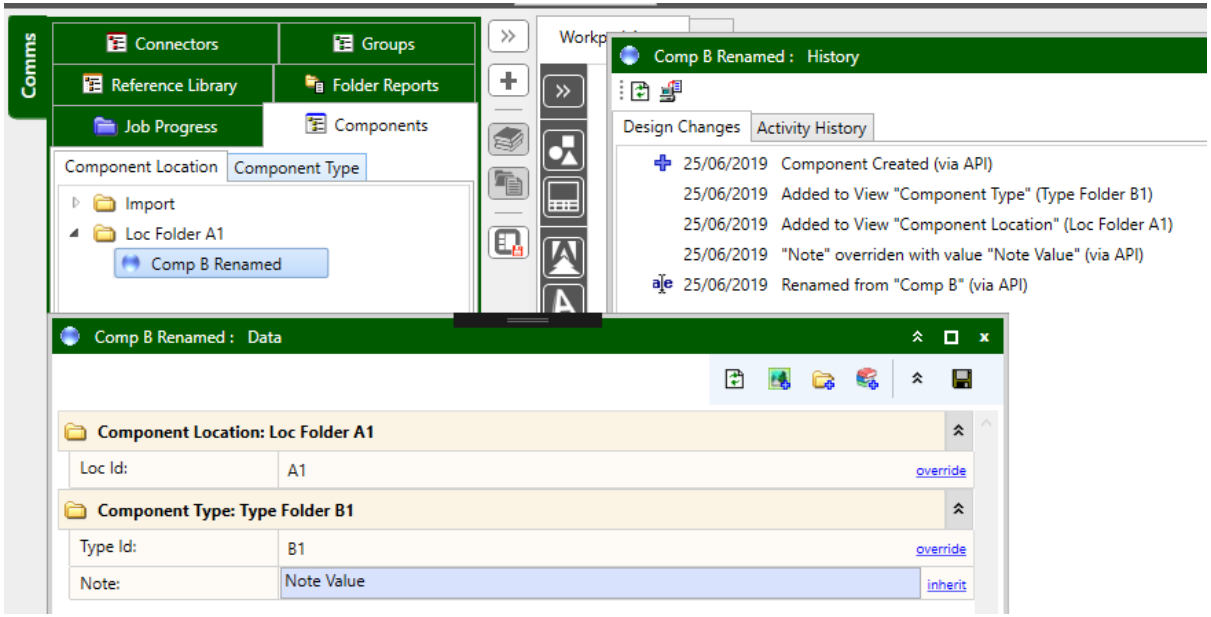
Existing Component Matching on Attribute then updating Tag

For this, we will match a component on the attribute name, then update its tag to the new value in TAG_Name

Header Data					
External_id	TAG_Name	DAD_Portal	DAD_Layer	MatchingAttribute_Name	MatchingAttribute_Value
124	Comp B Renamed	SIM	Comms	Note	Note Value

```
Insert into tblAPI_Import(External_id,TAG_Name,DAD_Portal,DAD_Layer,MatchingAttribute_Name,MatchingAttribute_Value)
Values ('124','Comp B Renamed','SIM','Comms','Note','Note Value')
```

This gives this component



This history log shows that it was renamed, and can also see from the results table can see that it matched existing then updated it's tag

```
select * from tblAPI_Import_Results where External_id = 124
```

DAD_UID	External_id	DAD_id	Date	Comment
7	124	145053	2019-06-25 11:05:00	Matched existing DAD component
7	124	145053	2019-06-25 11:05:00	Renamed

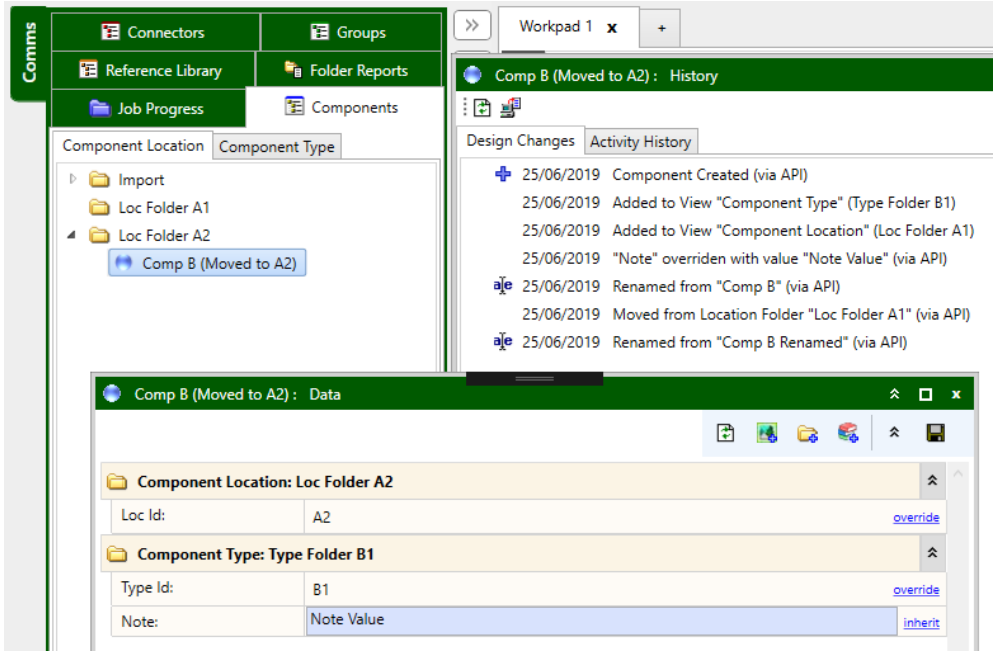
Existing Component matching on Tag, then changing Location parent and also updating tag

For this we'll match the previous component on the tag name. A Location tree folder has been created in DAD with the "Loc Id" attribute set to A2.

Header Data						Location/Type Data			
External_id	TAG_Name	DAD_Portal	DAD_Layer	MatchingAttribute_Name	MatchingAttribute_Value	LocationParentAttribute_Name	LocationParentAttribute_Value	TypeParentAttribute_Name	TypeParentAttribute_Value
125	Comp B (Moved to A2)	SIM	Comms	DAD:Tag Name	Comp B Renamed	Loc Id	A2	Type Id	B1

```
Insert into tblAPI_Import(External_id,TAG_Name,DAD_Portal,DAD_Layer,MatchingAttribute_Name,MatchingAttribute_Value,LocationParentAttribute_Name,LocationParentAttribute_Value,TypeParentAttribute_Name,TypeParentAttribute_Value) Values ('125','Comp B (Moved to A2)','SIM','Comms','DAD:Tag Name','Comp B Renamed','Loc Id','A2','Type Id','B1')
```

Moved and renamed the component as below



The results table shows the changes

```
select * from tblAPI_Import_Results where External_id = 125
```

DAD_UID	External_id	DAD_id	Date	Comment
12	125	145053	2019-06-25 11:30:00	Matched existing DAD component
12	125	145053	2019-06-25 11:30:00	Matched folder "Loc Folder A2" with Location Pare...
12	125	145053	2019-06-25 11:30:00	Moving to Location Folder "Loc Folder A2".
12	125	145053	2019-06-25 11:30:00	Matched folder "Type Folder B1" with Type Parent...
12	125	145053	2019-06-25 11:30:00	Renamed