

Using ETL to manage custom properties, network connections, and power measurements

(Version 2.0 by Greg Sterling – Added Power Measurement info)

General Overview

The most important item in the ETL import database is the configuration_item table. This table contains a list of all the assets the customer plans to manage using the ETL framework. Assets include rooms, rows, racks, racked and floor assets.

Sample configuration_item table data

ci_id	ci_description	ci_install_date	ci_location	ci_logic_name	ci_manufacturer	ci_model_name	ci_mounting	ci_name	ci_parent
1	Cisco Catalyst 1912	2016-09-26 17:14:52.352	14g/U-40/Rack-1/Test-Room-1/ETL-Test-Room/	Catalyst 1912	Cisco	Catalyst 1912	FRONT		bed3fc25
2	Dell PowerEdge R610	2016-09-26 17:18:05.997	14g/U-17/Rack-1/Test-Room-1/ETL-Test-Room/	PowerEdge R610	Dell	PowerEdge R610	FRONT		bed3fc25
3	Dell PowerEdge R610	2016-09-26 17:47:24.663	14g/U-22/Rack-1/Test-Room-1/ETL-Test-Room/	PowerEdge R610	Dell	PowerEdge R610	FRONT		bed3fc25
4								Global	
5			14g ETL-Test-Room/	Test-Room-1				Test-Room-1	daF04b51
6	APC Rack PDU, Switched, Zer	2016-09-12 09:25:55.536	14g Left-rear/Rack-1/Test-Room-1/ETL-Test-Room/	RFD0-B	APC	Rack PDU, Switched, Zero, 5 LEFT_REAR		RFD0-B	bed3fc25
7	APC Rack PDU, Switched, Zer	2016-09-12 09:25:52.683	14g Right-rear/Rack-1/Test-Room-1/ETL-Test-Room/	RFD0-A	APC	Rack PDU, Switched, Zero, 5 RIGHT_REAR		RFD0-A	bed3fc25
8	APC MetHelter 3x 42u 600	2016-09-12 09:25:21.718	14g Test-Room-1/ETL-Test-Room/	Rack-1	APC	MetHelter 3x 42u 600mm W		Rack-1	6cbb8894
9			14g					ETL-Test-Room	6b244c43

The ci_id field is the primary key for all data center assets. Each asset in DCO has a unique ci_id value. Existing data center asset information can be exported from DCO in the form of an ETL export database definition. The above sample is from an ETL export database.

To manage an asset using the ETL framework the item MUST have an entry in the configuration_item table an ETL import database and the item MUST be associated to an asset in DCO.

ETL Import database: configuration_item

To manage asset information from ETL, each asset must have an entry in the configuration_item table in an ETL import database. The ci_id field is the primary key so it should represent a unique asset id in the Import database. The ci_id can be user defined, it might be the unique asset id from another asset management system (i.e. rf-code, HP's ucmdb, ServiceNow, ...) or it can be the same ci_id value from the ETL export database.

ci_id	ci_description	ci_fingerprint	ci_ip_address	ci_location	ci_manufacturer	ci_model_name	ci_model_number	ci_name	ci_serial_number	state_flag	state_last_synchronized	state_last_updated	ci_id
1	Description		10.10.1.50	Dell	PowerEdge R610	R610	ETL-R610-Name	012345	new		14749246166595		1
2	Description		10.10.1.51	Dell	PowerEdge R610	R610	ETL-R610-Name-2	112345	new		14749246166648		1

The last field is the cit_id field value. This value defines the "type" of device which the configuration_item record represents. The number in this field must match an entry in the configuration_item_type table in the import database. There are over 30 item types in the table, but a few of the common cit_type values are:

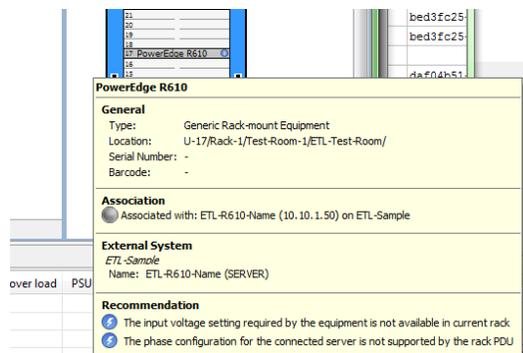
CIT_ID value	Type
1	Server
19	Power Distribution
20	Power Meter
23	Rack Power Distribution
29	UPS

The other field values in configuration_item are mostly optional. The other field values may be sourced from an external customer inventory management system.

The state_flag field should be set to “new” for new assets, “changed” for assets being updated, “deleted” for items being removed from the ETL database.

Assets new to DCO will appear in the “unassociated items” list (seen under the Planning -> Device-Association view). Assets may be associated manually or may be eligible for automatic association if the item properties (like ci_id, name, model, serial number) match existing assets in the opened DCO room.

Once an item is associated... customers may enable “external systems” (under tools->preferences->general->tool tips) in their tool-tip view to see the association data when mousing over an associated asset.

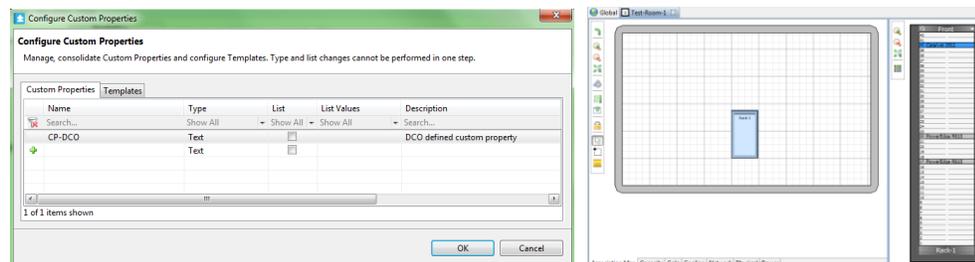


Using ETL to manage Custom Properties

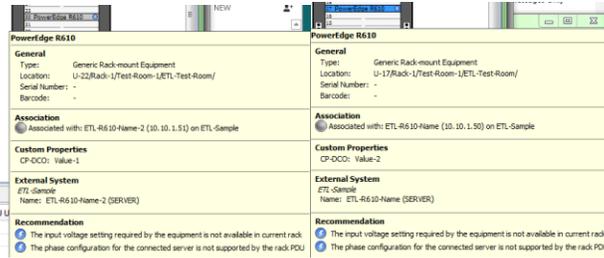
Custom properties may be managed using the ETL framework. Management includes the ability to add, change and remove custom properties assigned to DCO assets.

To manage custom properties for a DCO asset the item MUST first be defined in the configuration_item table in the ETL import database and the item MUST be associated to an asset in DCO.

Sample data. DCO 8.x server with a rack, two Dell servers, and one network switch. Server has a single defined custom properties called “CP-DCO”.



The custom property has been pre-assigned to both servers in the rack using the DCO client. The property is set to “Value-1” for one server, “Value-2” for the other.



Create an ETL based custom property

An entry for the target DCO asset must exist in the configuration_item table in the ETL import database.

Add an entry to configuration_item. In this example I used the ci_id value from my ETL export database for each of the Dell servers as the ci_id key value in my ETL import database. Using the ci_id value from DCO is not required, some users will generate a new key value, or they might re-use a key value from an external system which is feeding this data to DCO.

The ci_id, state_flag, and cit_id fields are all required. The other fields are optional. The cit_id field represents an integer value from the configuration_item_type table in the ETL export database. A value of "1" means the asset is a "server". The state_flag value should be set to "new" for new entries to this table.

Sample ETL import configuration_item table data.

ci_id	ci_description	ci_fingerprint	ci_ip_address	ci_location	ci_manufacturer	ci_model_name	ci_model_number	ci_name	ci_serial_number	state_flag	state_last_synchronized	state_last_updated	cit_id
[PK] character varying(255)	character varying(255)	character varying(255)	character varying(255)	character varying(255)	character varying(255)	character varying(255)	character varying(255)	character varying(255)	character varying(255)	character varying(16)	bigint	bigint	integer
1	1a434c34-1146-41dc-acba-408112e4d41e R610 Description		10.10.1.50		Dell	PowerEdge R610	R610	ETL-R610-Name	012345	new	1474924056595		1
2	4ed90d98-99ad-4a38-8613-b2f70b140bd4 R610 Description		10.10.1.51		Dell	PowerEdge R610	R610	ETL-R610-Name-2	112345	new	1474926614848		1

Once the configuration_item table entries are created. Those entries should be associated to existing items/entries in DCO.

Creating a new custom property entry in the configuration_item_property table

Add an entry to the configuration_item_property table. The ci_id field value should be set to the asset ci_id from the configuration_item table. The "state_flag" should be set to "new" for new custom properties, "changed" if you're changing a custom property. The "pt_id" field is also required. For custom properties this value has to be set to "2". For "tags" use "pt_id" set to "1". Reference the contents of the property_type table in the import database to see all of the entries.

	ci_id	ci_name	state_flag	state_last_synchronized	state_last_updated	ci_value	pt_id
	[PK] character varying(255)	[PK] character varying(255)	character varying(16)	character varying(16)	bigint	character varying(255)	integer
1	1a434c34-1146-41dc-acba-408112e4d41e	CP-ETL-1	new			Set-in-ETL	2
*							

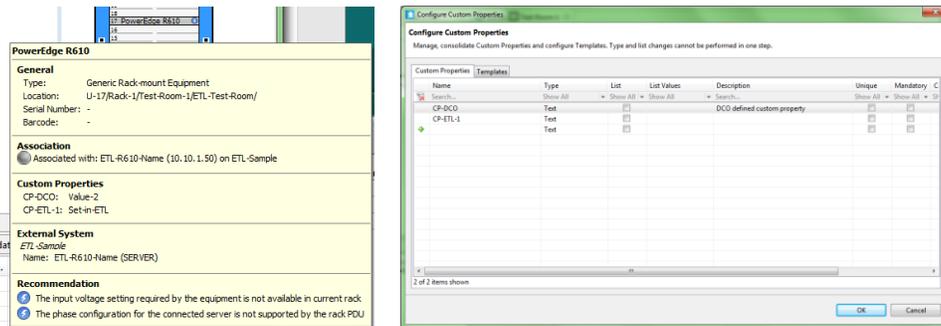
The above example creates a custom property called "CP-ETL-1", assigned to the asset which has the above ci_id value, and assigns the custom property value to "Set-in-ETL". The state_flag field is set to "new" because this is a new custom property.

	ci_id	ci_name	state_flag	state_last_synchronized	state_last_updated	ci_value	pt_id
	[PK] character varying(255)	[PK] character varying(255)	character varying(16)	character varying(16)	bigint	character varying(255)	integer
1	1a434c34-1146-41dc-acba-408112e4d41e	CP-ETL-1	synced	1476787686734		Set-in-ETL	2
*							

The state_flag will change to “synced” (see above screenshot) if the item is successfully sync’d to DCO.

Sync’ing to DCO occurs on the scheduled basis. The schedule is defined when the ETL definition is created (in the external systems tool). There is a “run now” button in the ETL definition tool in the DCO desktop client. When in “Planning – Device Association” mode, users may also right click on the ETL definition in the unassociated items list, and choose “synchronize external data” to force the ETL job to run.

The next time the DCO desktop client refreshes room data (to force a refresh of the desktop client, change the desktop client from planning mode to operations mode). Below is a screen shot of the asset tool-tip window and a screenshot of the custom property wizard which now lists the new custom property entry.

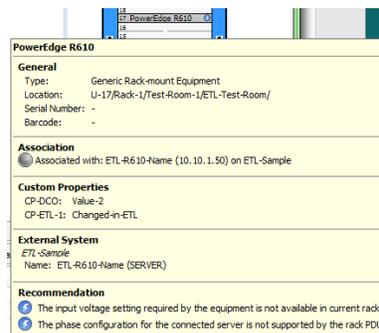


Changing an ETL custom property

Changing a custom property through ETL requires a change to the cip_value field in the appropriate record in the configuration_item_property table, and setting the “state_flag” field to “changed”. Customers can change setting specific records based on the asset ci_id or make wide-spread updates to records based on other criteria.

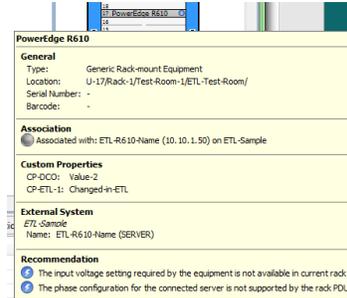
	ci_id [PK] character varying(255)	cip_name [PK] character varying(255)	state_flag character varying(16)	state_last_synchronized bigint	state_last_updated bigint	cip_value character varying(255)	pt_id integer
1	1a434c34-1146-41dc-acba-408112e4d41e	CP-ETL-1	changed	1476787686734		Changed-in-ETL	2
*							

Once re-sync’d with DCO, the custom property value will change.



Changing a DCO based Custom Property from ETL

It is also possible to use the ETL import database to change custom properties which were created from the DCO desktop client.

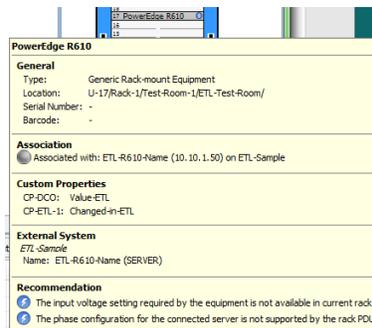


The above asset has an existing custom property called “CP-DCO” which was created and assigned to the asset using the DCO desktop client.

To use ETL to change this custom property. Add a record for the above asset to the configuration_item_property table, use the same custom property name “cip_name”, set the “state_flag” to “new”, and set the “cip_value” to the new custom property value. See the below example:

	ci_id [PK] character varying(255)	cip_name [PK] character varying(255)	state_flag character varying(16)	state_last_synchronized bigint	state_last_updated bigint	cip_value character varying(255)	pt_id integer
1	1a434c34-1146-41dc-acba-408112e4d41e	CP-ETL-1	synced	1476787911108		Changed-in-ETL	2
2	1a434c34-1146-41dc-acba-408112e4d41e	CP-DCO	new			Value-ETL	2
*							

Once sync'd with DCO, and the desktop client room info is refreshed, the existing custom property “CP-DCO” will be updated.



Delete an ETL based custom property

To delete a custom property from an asset, edit the appropriate record from the configuration_item_property table and set its “state_flag” to “deleted”.

	ci_id [PK] character varying(255)	cip_name [PK] character varying(255)	state_flag character varying(16)	state_last_synchronized bigint	state_last_updated bigint	cip_value character varying(255)	pt_id integer
1	1a434c34-1146-41dc-acba-408112e4d41e	CP-DCO	deleted	1476788512196		Value-ETL	2
2	1a434c34-1146-41dc-acba-408112e4d41e	CP-ETL-1	synced	1476788512196		Changed-in-ETL	2
*							

Setup Cable Types in cable_type table

First requirement for ETL based network connections is to setup entries in the “cable_type” table. This table is used to set cable color in ETL connection definitions.

	id [PK] character varying(255)	blue smallint	description character varying(255)	green smallint	red smallint
1	RJ45_LAN_ETL	0	RJ45 Cable	0	0
2	RJ45_MGMT_ETL	100	RJ45 ManageMentCabl	100	100
*					

Adding a network module to a device/switch/server

ETL can be used to define network modules in servers, switches, and other device types. To complete this task create a record in the port_configuration table which represents a network module for the asset represented by the “ci_id” value.

In the below example, I created a network module called “Motherboard_ETL” which has RJ45_REGULAR connections, 1 Gbps speed, and four connections. If this asset has additional network cards/modules you can create additional records.

	connector_type character varying(255)	module_name character varying(255)	start_port integer	port_count integer	port_speed character varying(255)	state_flag character varying(16)	state_last_synchronized bigint	state_last_updated bigint	ci_id [PK] character varying(255)
1	RJ45_REGULAR	Motherboard_ETL	1	4	1 Gbps	new	0	0	1a434c34-1146-41dc-acba-408112e4d41e
*									

Connector type field must match one of the predefined connector type entries from this page:

<https://dcimsupport.apc.com/display/DCIMDEVELOPER/ETL+Import+Database>

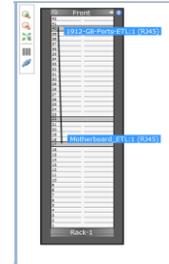
Once completed, sync the changes to DCO and refresh the DCO desktop client. The below screen shots displays separate entries to the port_configuration table adding network modules to a server and a network switch.

	connector_type character varying(255)	module_name character varying(255)	start_port integer	port_count integer	port_speed character varying(255)	state_flag character varying(16)	state_last_synchronized bigint	state_last_updated bigint	ci_id [PK] character varying(255)
1	RJ45_REGULAR	Motherboard_ETL	1	4	1 Gbps	synced	0	0	1a434c34-1146-41dc-acba-408112e4d41e
*									

	connector_type character varying(255)	module_name character varying(255)	start_port integer	port_count integer	port_speed character varying(255)	state_flag character varying(16)	state_last_synchronized bigint	state_last_updated bigint	ci_id [PK] character varying(255)
1	RJ45_REGULAR	Motherboard_ETL	1	4	1 Gbps	synced	0	0	1a434c34-1146-41dc-acba-408112e4d41e
2	RJ45_REGULAR	1912-GB-Ports-ETL	1	8	1 Gbps	new	0	0	0dfea3b2-af82-4de5-ad78-e3ecac719482
*									

The below screenshot displays the changes to the assets. The new network ports are visible on the server and the network switch.

Name	Serial Nu...	Barcode	Cable Type	Start Item	Start Port Name	Start Port	Start Port Type	End Item	End Port Name	End Port	End Port Type
Server-to-Switch	1234		RJ45 Cable	Catalyst 1912	1912-GB-Ports-ETL	1	RJ45	PowerEdge R610	Motherboard_ETL	1	RJ45

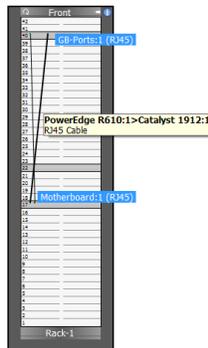


Create a network connection for network modules not already defined in ETL

You may also create network connections using modules which had been defined in the DCO desktop client (so they are not defined in the ETL import port_configuration table). In this case you must match the names of the existing network port modules.

id	bar desc	from_connector	from_module_name	from_item	from_port	from_port_type	name	part_nu	serial	state	flag	to_connector	to_module_name	to_cable_type_id	from_id	to_id				
1	Server-to-Switch-Port-1	Serv	RJ45_REGULAR	Motherboard_ETL	1	0	0	48	CableC	Cable-48	ServerToSwitch	C48	1234	synced	RJ45_REGULAR	1912-GB-Ports-ETL	1	RJ45_LAN_ETL	1a434c34-1146-41dc-acba-408112e4d41e	0dfea3b2-af02-4de5-ad78-e3eac719482
2	Server-to-Switch-Port-2	Serv	RJ45_REGULAR	Motherboard	1	0	0	48	CableC	Cable-48	ServerToSwitch-2	C48	1234	synced	RJ45_REGULAR	GB-Ports	1	RJ45_LAN_ETL	1a434c34-1146-41dc-acba-408112e4d41e	0dfea3b2-af02-4de5-ad78-e3eac719482

Once sync'd to DCO and the desktop client room data is refreshed the new port connection is visible.



Name	Serial Nu...	Barcode	Cable Type	Start Item	Start Port Name	Start Port	Start Port Type	End Item	End Port Name	End Port	End Port Type
ServerToSwitch	1234		RJ45 Cable	Catalyst 1912	1912-GB-Ports-ETL	1	RJ45	PowerEdge R610	Motherboard_ETL	1	RJ45
ServerToSwitch-2	1234		RJ45 Cable	Catalyst 1912	GB-Ports	1	RJ45	PowerEdge R610	Motherboard	1	RJ45

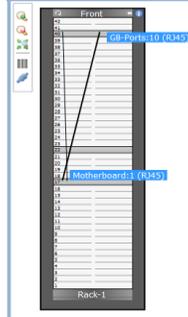
Changing a network connection

Existing ETL based network connections can be modified by editing the network connection entry and setting the "state_flag" to "changed". In this case the to_port was changed from "1" to "10".

id	bar desc	from_connector	from_module_name	from_item	from_port	from_port_type	name	part_nu	serial	state	flag	to_connector	to_module_name	to_cable_type_id	from_id	to_id				
1	Server-to-Switch-Port-1	Serv	RJ45_REGULAR	Motherboard_ETL	1	0	0	48	CableC	Cable-48	ServerToSwitch	C48	1234	synced	RJ45_REGULAR	1912-GB-Ports-ETL	1	RJ45_LAN_ETL	1a434c34-1146-41dc-acba-408112e4d41e	0dfea3b2-af02-4de5-ad78-e3eac719482
2	Server-to-Switch-Port-2	Serv	RJ45_REGULAR	Motherboard	1	0	0	48	CableC	Cable-48	ServerToSwitch-2	C48	1234	changed	RJ45_REGULAR	GB-Ports	10	RJ45_LAN_ETL	1a434c34-1146-41dc-acba-408112e4d41e	0dfea3b2-af02-4de5-ad78-e3eac719482

Network connection record after the record is sync'd to DCO.

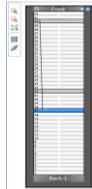
id	bar desc	from_connector	from_module_name	from_item	from_port	from_port_type	name	part_nu	serial	state	flag	to_connector	to_module_name	to_cable_type_id	from_id	to_id				
1	Server-to-Switch-Port-1	Serv	RJ45_REGULAR	Motherboard_ETL	1	0	0	48	CableC	Cable-48	ServerToSwitch	C48	1234	synced	RJ45_REGULAR	1912-GB-Ports-ETL	1	RJ45_LAN_ETL	1a434c34-1146-41dc-acba-408112e4d41e	0dfea3b2-af02-4de5-ad78-e3eac719482
2	Server-to-Switch-Port-2	Serv	RJ45_REGULAR	Motherboard	1	0	0	48	CableC	Cable-48	ServerToSwitch-2	C48	1234	synced	RJ45_REGULAR	GB-Ports	10	RJ45_LAN_ETL	1a434c34-1146-41dc-acba-408112e4d41e	0dfea3b2-af02-4de5-ad78-e3eac719482



Delete a connection

Existing ETL based network connections can be deleted by editing the network connection entry and setting the “state_flag” to “deleted”. Once sync’d to DCO and the room info is refreshed in the DCO desktop client the network connection will disappear.

id	hw_desc	from_connector	from_module_name	from_port	is_in	is_b	length	manuf	model_name	name	part_no	serial	state	st	st_i	to_connector	to_module_name	to_cable_type_id	from_id	to_id	
1	Server-to-Switch-Port-1	SrvR245_REGULAR	Motherboard_ETL	1	0	0	48	CableC	Cable-48	ServerToSwitch	C48	1234	synced			R245_REGULAR	GB-Ports-ETL	1	R245_LAB_ETL	1a434c34-1146-41dc-80ba-408112e4541e	05fea3b2-affe-4de5-ed78-e3e0ac719482
2	Server-to-Switch-Port-2	SrvR245_REGULAR	Motherboard	1	0	0	48	CableC	Cable-48	ServerToSwitch-2	C48	1234	deleted			R245_REGULAR	GB-Ports	10	R245_LAB_ETL	1a434c34-1146-41dc-80ba-408112e4541e	05fea3b2-affe-4de5-ed78-e3e0ac719482



Adding Power Measurements to DCO using an ETL import database

Power measurements and environment values may be managed using the ETL framework. For power values measurement data may be in the form of watt, Kw, and amps. For environmental data, temperature and humidity may be specified.

First task is to verify you have created entries in configuration_item for each asset you plan to add power measurements. In the below screenshot I have added a configuration item record for a UPS.

ci_id	ci_description	ci_fingerprint	ci_ip_address	ci_location	ci_manufacturer	ci_model_name	ci_model_number	ci_name	ci_serial_number	state_flag	state_last_synchronized	state_last_u...	ci_id
001	Test UPS - 1	(null)	10.0.0.1	(null)	ACME	UPS	1	UPS_A	12345	synced	1485406340974	(null)	29

The ci_id value must be unique in the import database, if you have the ci_id value of the actual UPS which already exists in DCO, then you can use that ci_id value in the import database. This will allow the device to be auto-associated to DCO later on. Note for a UPS device the ci_id value in the last column has been set to 29. Also note, the “state_flag” in the above screenshot is set to “synced”, for new records you should set this value to “new”.

A SQL insert statement like this would add the above record:

```
INSERT INTO configuration_item VALUES ('001', 'Test UPS - 1', NULL, '10.0.0.1', NULL, 'ACME', 'UPS', '1', 'UPS_A', '12345', 'new', 1485406340974, NULL, 29);
```

Power measurements for DCO assets excluding breaker panels should be added to the configuration_item_measurement table. (Breaker panel measurements can be added to a separate table which I'll describe further below).

I have pasted a sample set of power measurements for the UPS we previously added to the configuration_item table.

*	cim_id	cim_interval_endtime	cim_interval_starttime	state_flag	state_last_synchronized	state_last_updated	cim_subid	cim_value	mc_id	mt_id	mu_id	ci_id
1	UPS_avg_11	1485272426000	1485062426000	synced	1523008801737		0 1	11.0	11	3	2.001	
2	UPS_avg_12	1485272426000	1485062426000	synced	1523008801735		0 1	12.0	12	3	2.001	
3	UPS_avg_13	1485272426000	1485062426000	synced	1523008801734		0 1	13.0	13	3	2.001	
4	UPS_peak_11	1485272426000	1485062426000	synced	1523008801734		0 1	22.0	11	2	2.001	
5	UPS_peak_12	1485272426000	1485062426000	synced	1523008801734		0 1	24.0	12	2	2.001	
6	UPS_peak_13	1485272426000	1485062426000	synced	1523008801737		0 1	26.0	13	2	2.001	

The cim_id column is a unique name for the measurement.

The cim_interval_endtime and cim_interval_starttime values represent the time range for which the measurement was taken. Those values represent the number of milliseconds since the epoch time. Web sites like <https://www.epochconverter.com/> can be used to translate these values. The cim_interval_starttime value in the above records represents "Sunday, January 22, 2017 5:20:26 AM"

state_flag should be set to "new" for new measurement values, "changed" if you are changing a previous value, and "deleted" if you are removing power measurements.

cim_value represents the measurement value.

mc_id represents an entry from the measurement_class table. The table has a row for a number of classes including "CURRENT", "POWER", "POWER_L1", "POWER_L2" and so on.

mt_id represents an entry from the measurement_type table. This entry represents a description of the measurement value including "minimum", "maximum", "average", "peak_day".

mu_id represents an entry from the measurement_unit table. This entry represents the measurement unit of measure. Units include "W", "kW", "A", "F", "C" and so on.

The last field, ci_id must match a ci_id entry in the configuration_item table.

So if we look at the top row in the above measurement records

*	cim_id	cim_interval_endtime	cim_interval_starttime	state_flag	state_last_synchronized	state_last_updated	cim_subid	cim_value	mc_id	mt_id	mu_id	ci_id
1	UPS_avg_11	1485272426000	1485062426000	synced	1523008801737		0 1	11.0	11	3	2.001	

This measurement would be called UPS_avg_11. The cim_value is 11, the units (mu_id) is kW, the type (mt_id) says the value is an AVERAGE, and the class (mc_id) is POWER_L1 (meaning this is phase 1 power). Note, the above record has a state_flag set to "synced" because this data has already been consumed by DCO. If you are adding new records your state_flag record should be set to "new".

Sample SQL statements to insert the above measurements might look like this:

```
INSERT INTO configuration_item_measurement VALUES ('UPS_peak_11', 1485272426000, 1485062426000, 'new', 1493049601710, 0, '1', 22, 11, 2, 2, '001');
```

```
INSERT INTO configuration_item_measurement VALUES ('UPS_peak_12', 1485272426000, 1485062426000, 'new', 1493049601710, 0, '1', 24, 12, 2, 2, '001');
```

```
INSERT INTO configuration_item_measurement VALUES ('UPS_peak_13', 1485272426000, 1485062426000, 'new', 1493049601709, 0, '1', 26, 13, 2, 2, '001');
```

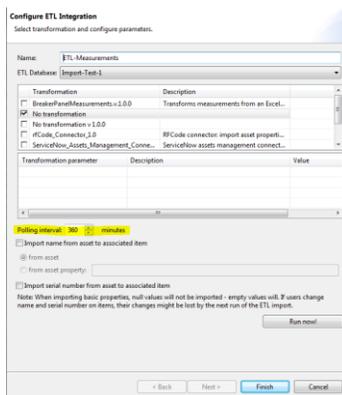
```
INSERT INTO configuration_item_measurement VALUES ('UPS_avg_11', 1485272426000, 1485062426000, 'new', 1493049601710, 0, '1', 11, 11, 3, 2, '001');
```

```
INSERT INTO configuration_item_measurement VALUES ('UPS_avg_12', 1485272426000, 1485062426000, 'new', 1493049601710, 0, '1', 12, 12, 3, 2, '001');
```

```
INSERT INTO configuration_item_measurement VALUES ('UPS_avg_13', 1485272426000, 1485062426000, 'new', 1493049601710, 0, '1', 13, 13, 3, 2, '001');
```

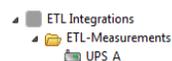
I recommend adding peak and average power measurements for each item. If you do not have one or the other measurement (i.e. you only have peak power values), then I'd recommend setting the other value to be the same. In the above example, if you only have peak power values, set average and peak power values to be the same.

ETL import records are consumed by DCO based on the schedule defined in the external systems configuration entry for the ETL job. In the below task the job runs every 6 hours (360 minutes).

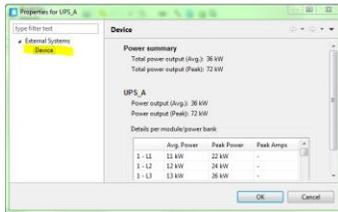


Also note, the “Run now!” button tells DCO to consume the data immediately. When “Run now!” is clicked the task is scheduled to run immediately, but may take a little time to start if DCO is already executing other ETL tasks.

Unassociated ETL import devices will appear under the “ETL Integrations” section of the unassociated devices view (view is seen from Planning -> Device Association).

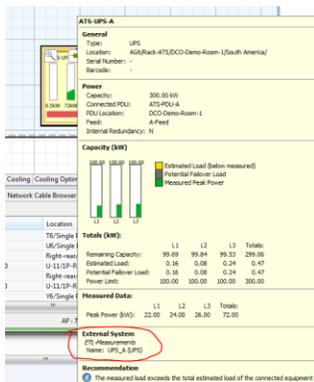


If the ETL entries are properly setup you should see the power data in the device properties for this item.



To see the power information on the DCO asset (i.e. in the power overlay) then the unassociated device must be associated to a DCO asset. To do that, drag the unassociated entry from the list on top of the DCO device.

If the record in the configuration_item table for this device has an assigned ci_id which matches an actual UPS which is already in this room, then DCO can automatically associated the UPS to the device in the room.



Adding power measurements to a breaker panel

The same ETL import database can be used to add power measurements to a breaker panel. Table breaker_panel_measurement is used for this purpose. DCO supports use of the xls spreadsheet which automatically populates this table with appropriate information.

Like the configuration_item_measurement table this table takes a series of measurements which have unique names (id columns) but since this data is meant for breaker panels there are fewer fields to fill-in since breaker poles only support specific measurement types.

#	id	amps	circuit_number	panel_id	power_phase	state_flag	state_last_synchronized	state_last_updated	time	watts	ci_id
1	4b37ae389496276869415657e59b74739ea56d46	(null)	1	Panel-A	L1	synced	1486477250367	(null)	1485287266000	197.0	009ba80f-db61-469e-bea4-7d166c8f90f1
2	4a7e507344514ef7c90b2763e03fd13ad6d0aaff	(null)	2	Panel-A	L1	synced	1486477250367	(null)	1485287266000	230.0	009ba80f-db61-469e-bea4-7d166c8f90f1
3	c2c20c74219a62be4eb8fcd0ceb7248378dc65f	(null)	3	Panel-A	L2	synced	1486477250367	(null)	1485287266000	368.0	009ba80f-db61-469e-bea4-7d166c8f90f1
4	fd21c3e967cb8e2c5070b252078118ae49edbe2	(null)	4	Panel-A	L2	synced	1486477250367	(null)	1485287266000	402.0	009ba80f-db61-469e-bea4-7d166c8f90f1
5	c4ca9d806f52f5784e4478f1235a71df015ed7d	(null)	5	Panel-A	L3	synced	1486477250367	(null)	1485287266000	242.0	009ba80f-db61-469e-bea4-7d166c8f90f1
6	1fd8b2e8c585fb08958ae09a568cfe157bb04c9e	(null)	6	Panel-A	L3	synced	1486477250367	(null)	1485287266000	303.0	009ba80f-db61-469e-bea4-7d166c8f90f1
7	364011452b03d25b1f4b12856b9e30d94271587e	(null)	7	Panel-A	L1	synced	1486477250367	(null)	1485287266000	295.0	009ba80f-db61-469e-bea4-7d166c8f90f1
8	36c21c9fe1706fcd0d5112cce8ba59ada9ada823	(null)	8	Panel-A	L1	synced	1486477250367	(null)	1485287266000	333.0	009ba80f-db61-469e-bea4-7d166c8f90f1
9	de58ae1f274a4c3c84397fdd54ef5495000eb2	(null)	9	Panel-A	L2	synced	1486477250367	(null)	1485287266000	338.0	009ba80f-db61-469e-bea4-7d166c8f90f1

The "id" column is a unique name for the measurement. In the above example it's the unique id of the breaker with the timestamp appended to it to make it unique.

Fields “amps” or “watts” will contain the power measurements in either amps or watts units (not it is not kW).

Field “circuit_number” is the breaker panel circuit number.

Field “panel_id” is the name of the panel_id in the DCO asset. This name needs to match.

Field “time” is the timestamp from which the measurement was taken.

Field “ci_id” must match the ci_id of an entry in the configuration_item table.

#	id	amps	circuit_number	panel_id	power_phase	state_flag	state_last_synchronized	state_last_updated	time	watts	ci_id
1	4b37ae389496276869415657e59b74739ea56d46 1485287266000	(null)	1	Panel-A	L1	syncd	1486477250367	(null)	1485287266000	197.0	009ba80f-db61-469e-bee4-7d166c8f90f1
2	4a7e507344514ef7c90b276e03fd13ad6d0aaff 1485287266000	(null)	2	Panel-A	L1	syncd	1486477250367	(null)	1485287266000	230.0	009ba80f-db61-469e-bee4-7d166c8f90f1
3	c2c20c74219a62bed4eb8cfd0ceb7248378dc65f 1485287266000	(null)	3	Panel-A	L2	syncd	1486477250367	(null)	1485287266000	368.0	009ba80f-db61-469e-bee4-7d166c8f90f1
4	fd21c3e967cb8e2c5070b252078118aef49edbe2 1485287266000	(null)	4	Panel-A	L2	syncd	1486477250367	(null)	1485287266000	402.0	009ba80f-db61-469e-bee4-7d166c8f90f1
5	c4ca9d806f52f5784e4478f1235a71d1f015ed7d 1485287266000	(null)	5	Panel-A	L3	syncd	1486477250367	(null)	1485287266000	242.0	009ba80f-db61-469e-bee4-7d166c8f90f1
6	1fd8b2e8c585fb08958ae09a568cfe157bbb4c9a 1485287266000	(null)	6	Panel-A	L3	syncd	1486477250367	(null)	1485287266000	303.0	009ba80f-db61-469e-bee4-7d166c8f90f1
7	364011452b03d25b1f4b12856b9e30d9421587e 1485287266000	(null)	7	Panel-A	L1	syncd	1486477250367	(null)	1485287266000	295.0	009ba80f-db61-469e-bee4-7d166c8f90f1
8	36c21c9fe1706fc0d5112cce8ba59ada9ada8f823 1485287266000	(null)	8	Panel-A	L1	syncd	1486477250367	(null)	1485287266000	333.0	009ba80f-db61-469e-bee4-7d166c8f90f1
9	de58a61f274a4c3df84397f6db54ef5495000eb2 1485287266000	(null)	9	Panel-A	L2	syncd	1486477250367	(null)	1485287266000	338.0	009ba80f-db61-469e-bee4-7d166c8f90f1