

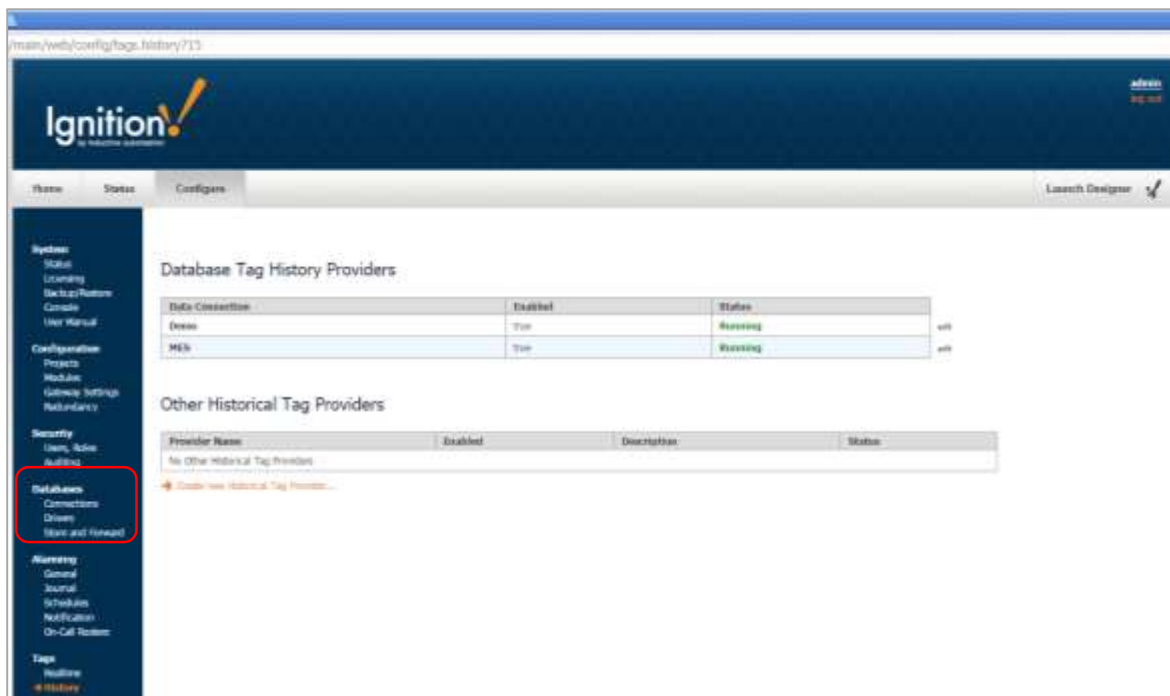
Dream Report Tech Note – December 12, 2014

Using Dream Report with Ignition® Data Logging

Ignition® SCADA by Inductive Automation provides a few different ways to log process data into SQL database. One of the methods is to simply select tags to be historized for trending, and they will be logged to a set of “generic” tables in the project’s database. Another method is to create “Transaction” groups in Ignition Designer, where you select the items/tags to be logged, storage rate, and the specific SQL table into which the data will be logged. Dream Report can access either of these data table types, but with slightly different approaches.

I. Tag History Logging

In the Ignition configuration environment, the project database is configured by creating a connection to a specific database, on a specific database platform (MySQL, MS SQL Server, etc.)



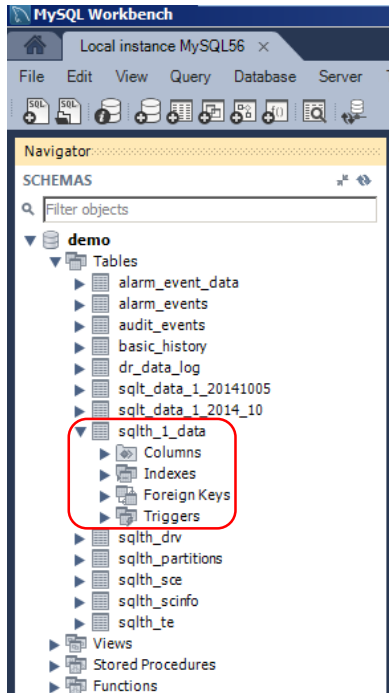
Then, under the “Tags – History” configuration, you can specify how the data is “partitioned” – i.e., the mechanism by which new tables are created periodically in which the data will be logged:

Edit Database Tag History Provider

Main	
Data Connection	Demo
Enabled	<input checked="" type="checkbox"/> Enable Tag History on this data connection (default: true)
Data Partitioning	
Enable Partitioning	<input checked="" type="checkbox"/> The built-in partitioning system breaks up data into separate tables of a specified time frame. This can improve performance and make certain maintenance tasks easier. (default: true)
Partition Length	1 How big a data partition should be (in time, with the units below) (default: 1)
Partition Units	MONTH (default: MONTH)
Enable Pre-processed Partitions	<input checked="" type="checkbox"/> Pre-processed partitions will use more space in the database, but can improve query speed by summarizing data, reducing the amount that must be loaded. (default: false)
Pre-processed Window Size (seconds)	60 When pre-processing is turned on, the data will be summarized into blocks of this size. (default: 60)

The default data partition length is set for 1 Month. What this means is that every month, a new table will be created in the SQL database, and the tag history will be logged to that month's table. From a trending (or query) efficiency standpoint, smaller tables of data can be queried, allowing for faster data retrieval. The downside to the data partitioning is that from reporting standpoint, it is difficult to know which table name(s) to query. To address this, you can configure a very long data partition – say, 10 years – or, simply not check the “Enable Data Partitioning” option. This is preferable for reporting, as you will always have a specific set of tables to work with, but may result in slower historical trend updates in Ignition.

Once the historical logging has been configured in Ignition, the project's database will get created, with the necessary tables. The image below shows a database for the Ignition “Demo” project, created in the MySQL database system. With no data partitioning configured, a single table, named *sqlth_1_data* below, is created, and all data will be logged to that table. If partitioning were enabled, multiple tables (named like *sql_data_1_2014_10* below – containing data for the month of Oct, 2014) would be created. We will assume that a single table will be used.



Looking at the contents of the table, we see that the data is not easily understood at first glance:

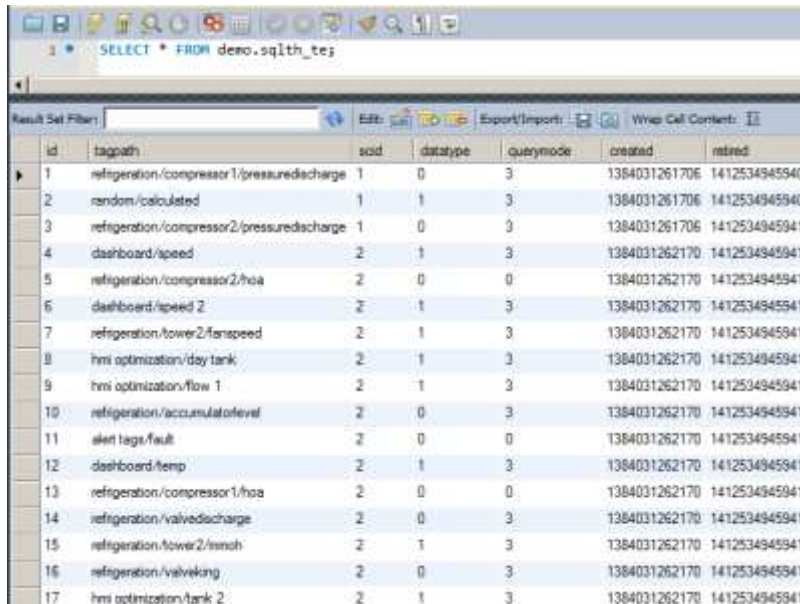
The screenshot shows the MySQL Workbench query editor with the query `SELECT * FROM demo.sqlth_1_data;` and the resulting data set. The data is presented in a table with the following columns: tagid, intvalue, floatvalue, stringvalue, datevalue, dataintegrity, and t_stamp.

tagid	intvalue	floatvalue	stringvalue	datevalue	dataintegrity	t_stamp
39	NULL	2000	NULL	NULL	192	1418236856471
39	NULL	1928.55102...	NULL	NULL	192	1418236866471
39	NULL	1616.64807...	NULL	NULL	192	1418236876471
39	NULL	1896.59973...	NULL	NULL	192	1418236886471
40	64	NULL	NULL	NULL	192	1412697654851
40	61	NULL	NULL	NULL	192	1412697664850
40	74	NULL	NULL	NULL	192	1412697674852
40	71	NULL	NULL	NULL	192	1412697684866
40	60	NULL	NULL	NULL	192	1412697694852
40	52	NULL	NULL	NULL	192	1412697704858
40	29	NULL	NULL	NULL	192	1412697714850
40	17	NULL	NULL	NULL	192	1412697724851
40	1	NULL	NULL	NULL	192	1412697734850
40	15	NULL	NULL	NULL	192	1412697744851
40	15	NULL	NULL	NULL	192	1412697754850
40	0	NULL	NULL	NULL	192	1412697764850
40	4	NULL	NULL	NULL	192	1412697774850
40	12	NULL	NULL	NULL	192	1412697784851
40	12	NULL	NULL	NULL	192	1412697794851
40	0	NULL	NULL	NULL	192	1412697804850

A few things stand out here, which will pose a challenge for reporting on the data:

- Tagnames are not logged with the data values, only a numeric *tagid*
- There are separate fields for integer, float, and string data types
- The timestamp is shown in Unix time format

This table separate table will need to be joined with the *sqlth_te* table to cross-reference the tag ID's with the actual tagnames:



id	tagpath	scid	datatype	querymode	created	retired
1	refrigeration/compressor1/pressuredischage	1	0	3	1384031261706	1412534945940
2	random/calculated	1	1	3	1384031261706	1412534945940
3	refrigeration/compressor2/pressuredischage	1	0	3	1384031261706	1412534945941
4	dashboard/speed	2	1	3	1384031262170	1412534945941
5	refrigeration/compressor2/hoa	2	0	0	1384031262170	1412534945941
6	dashboad/speed 2	2	1	3	1384031262170	1412534945941
7	refrigeration/tower2/fanspeed	2	1	3	1384031262170	1412534945941
8	hmi optimization/day tank	2	1	3	1384031262170	1412534945941
9	hmi optimization/flow 1	2	1	3	1384031262170	1412534945941
10	refrigeration/accumulatorlevel	2	0	3	1384031262170	1412534945941
11	sket tags/fault	2	0	0	1384031262170	1412534945941
12	dashboad/temp	2	1	3	1384031262170	1412534945941
13	refrigeration/compressor1/hoa	2	0	0	1384031262170	1412534945941
14	refrigeration/valvedischage	2	0	3	1384031262170	1412534945941
15	refrigeration/tower2/minch	2	1	3	1384031262170	1412534945941
16	refrigeration/valveking	2	0	3	1384031262170	1412534945941
17	hmi optimization/tank 2	2	1	3	1384031262170	1412534945941

Fortunately, we can create a SQL “view” to organize the data into a human-readable form. We actually need to create two views – one to display the numeric data (integer, float and discrete data points, and one for string tags.) The two views are scripted at follows, but please note that the syntax will vary slightly from one database package to another:

View: v_ignitiondata

```
CREATE
ALGORITHM = UNDEFINED
DEFINER = `root`@`localhost`
SQL SECURITY DEFINER
VIEW `v_ignitiondata` AS
select
    from_unixtime((`d`.`t_stamp` / 1000)) AS `DateTime`,
    `t`.`tagpath` AS `tagpath`,
    `d`.`intvalue` AS `Value`
from
    (`sqlth_1_data` `d`
    join `sqlth_te` `t` ON ((`d`.`tagid` = `t`.`id`)))
where
```

```

        (`d`.`intvalue` is not null)
union select
    from_unixtime((`d`.`t_stamp` / 1000)) AS `DateTime`,
    `t`.`tagpath` AS `tagpath`,
    `d`.`floatvalue` AS `Value`
from
    (`sqlth_1_data` `d`
    join `sqlth_te` `t` ON ((`d`.`tagid` = `t`.`id`)))
where
    (`d`.`floatvalue` is not null)
order by `DateTime`

```

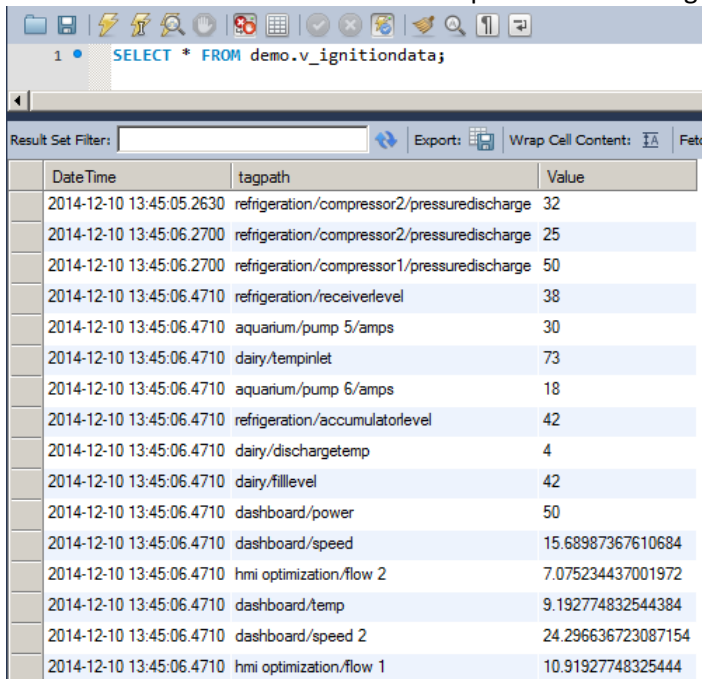
View: v_ignitiondata_strings

```

CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `v_ignitiondata_strings` AS
select
    from_unixtime((`d`.`t_stamp` / 1000)) AS `DateTime`,
    `t`.`tagpath` AS `tagpath`,
    `d`.`stringvalue` AS `Value`
from
    (`sqlth_1_data` `d`
    join `sqlth_te` `t` ON ((`d`.`tagid` = `t`.`id`)))
where
    (`d`.`stringvalue` is not null)
order by `DateTime`

```

When we query the *v_ignitiondata* view, for instance, we see data nicely displayed for reporting. We will reference this view from Dream Report when creating reports:



DateTime	tagpath	Value
2014-12-10 13:45:05.2630	refrigeration/compressor2/pressuredischarge	32
2014-12-10 13:45:06.2700	refrigeration/compressor2/pressuredischarge	25
2014-12-10 13:45:06.2700	refrigeration/compressor1/pressuredischarge	50
2014-12-10 13:45:06.4710	refrigeration/receiverlevel	38
2014-12-10 13:45:06.4710	aquarium/pump 5/amps	30
2014-12-10 13:45:06.4710	dairy/tempinlet	73
2014-12-10 13:45:06.4710	aquarium/pump 6/amps	18
2014-12-10 13:45:06.4710	refrigeration/accumulatorlevel	42
2014-12-10 13:45:06.4710	dairy/disargetemp	4
2014-12-10 13:45:06.4710	dairy/filllevel	42
2014-12-10 13:45:06.4710	dashboard/power	50
2014-12-10 13:45:06.4710	dashboard/speed	15.68987367610684
2014-12-10 13:45:06.4710	hmi optimization/flow 2	7.075234437001972
2014-12-10 13:45:06.4710	dashboard/temp	9.192774832544384
2014-12-10 13:45:06.4710	dashboard/speed 2	24.296636723087154
2014-12-10 13:45:06.4710	hmi optimization/flow 1	10.91927748325444

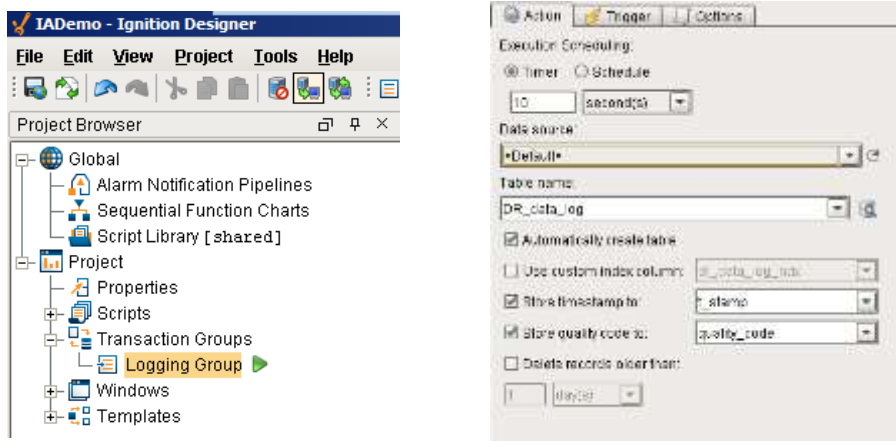
Note: For more information configuring Ignition data logging, please refer to the Ignition product documentation.

II. Transaction Groups

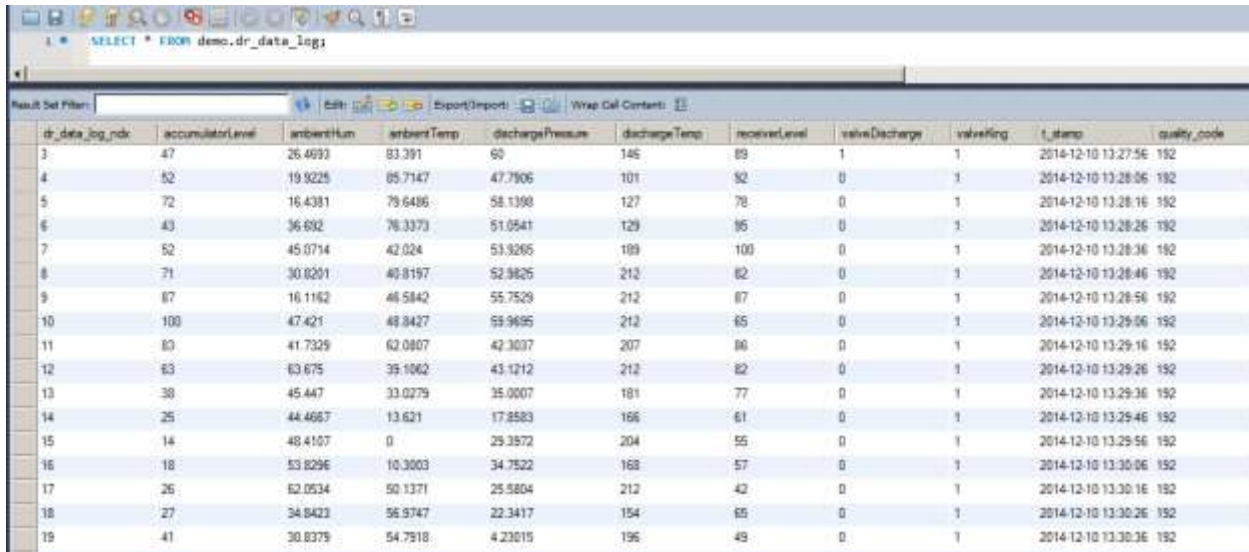
Transaction Groups are created in Ignition Designer:



In the example below, we created a transaction group named “*Logging Group*”, selected a set of tags to include in that group, and set the storage rate to be every 10 seconds. Additionally, we specified the table name (into which the data will be logged) as “*Dream Report_data_log*”.



The resulting table created in the (MySQL) database is easily understood, showing a field for each logged tag, and a timestamp field showing the local date and time:



dr_data_log_nbr	accumulatorLevel	ambientHum	ambientTemp	dischargePressure	dischargeTemp	receiverLevel	valveDischarge	valveFlg	t_stamp	quality_code
3	47	26.4693	83.391	60	146	89	1	1	2014-12-10 13:27:56	192
4	52	19.9225	85.7147	47.7906	101	92	0	1	2014-12-10 13:28:06	192
5	72	16.4381	78.6486	58.1398	127	78	0	1	2014-12-10 13:28:16	192
6	43	36.632	76.3373	51.0541	129	96	0	1	2014-12-10 13:28:26	192
7	52	45.0714	42.024	53.3285	109	100	0	1	2014-12-10 13:28:36	192
8	71	30.8201	40.8197	52.9825	212	82	0	1	2014-12-10 13:28:46	192
9	87	16.1162	46.5842	55.7529	212	87	0	1	2014-12-10 13:28:56	192
10	108	47.421	48.8427	59.9695	212	65	0	1	2014-12-10 13:29:06	192
11	80	41.7329	62.0807	42.3037	207	86	0	1	2014-12-10 13:29:16	192
12	63	63.675	39.1062	43.1212	212	82	0	1	2014-12-10 13:29:26	192
13	38	45.447	33.0279	35.0007	181	77	0	1	2014-12-10 13:29:36	192
14	25	44.4667	13.621	17.8583	166	61	0	1	2014-12-10 13:29:46	192
15	14	48.4107	0	29.3572	204	55	0	1	2014-12-10 13:29:56	192
16	18	53.8296	10.3003	34.7522	168	57	0	1	2014-12-10 13:30:06	192
17	26	62.0534	50.1371	25.5804	212	42	0	1	2014-12-10 13:30:16	192
18	27	34.8423	56.9747	22.3417	154	65	0	1	2014-12-10 13:30:26	192
19	41	38.8379	54.7918	4.23015	196	49	0	1	2014-12-10 13:30:36	192

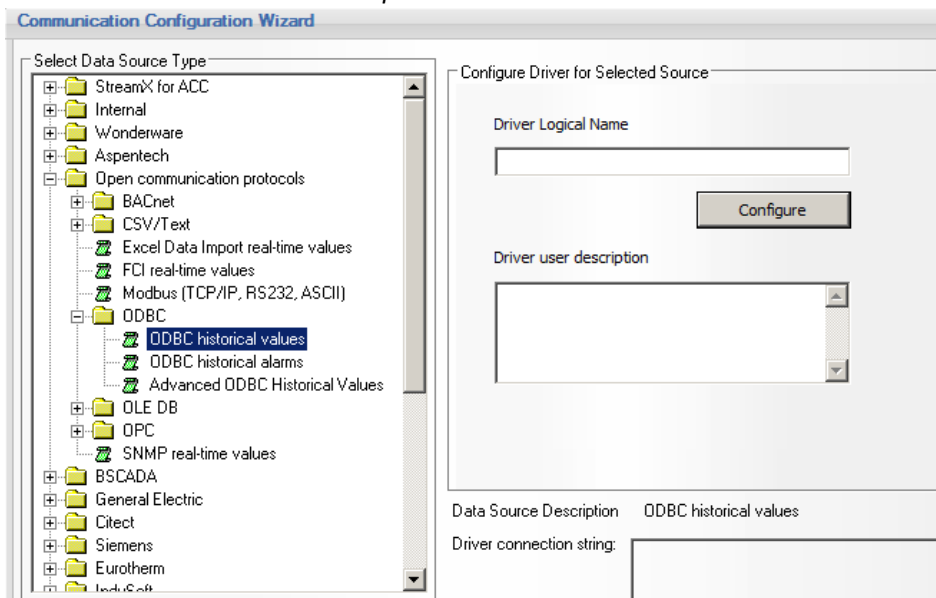
We're now ready to create reports in Dream Report with the Ignition-logged data...

Dream Report Communication Driver Configuration

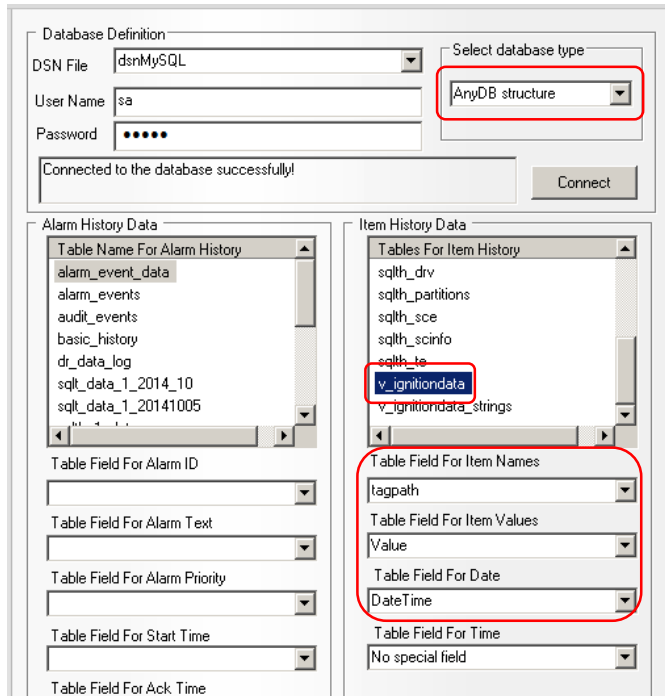
In Dream Report we will use the ODBC Communication Driver, to connect the Ignition SQL database, and then will connect to either the SQL view(s) or transaction group tables in the database. The following steps assume you have first used the Windows ODBC Manager (32-bit) to create an ODBC DSN (Data Source Name) to the specific Ignition database being used.

To configure the connections:

1. Open the "Communication Configuration Wizard" in Dream Report Studio, and select the "ODBC Historical Values" driver from *Open Communication Protocols > ODBC > ODBC Historical Values*

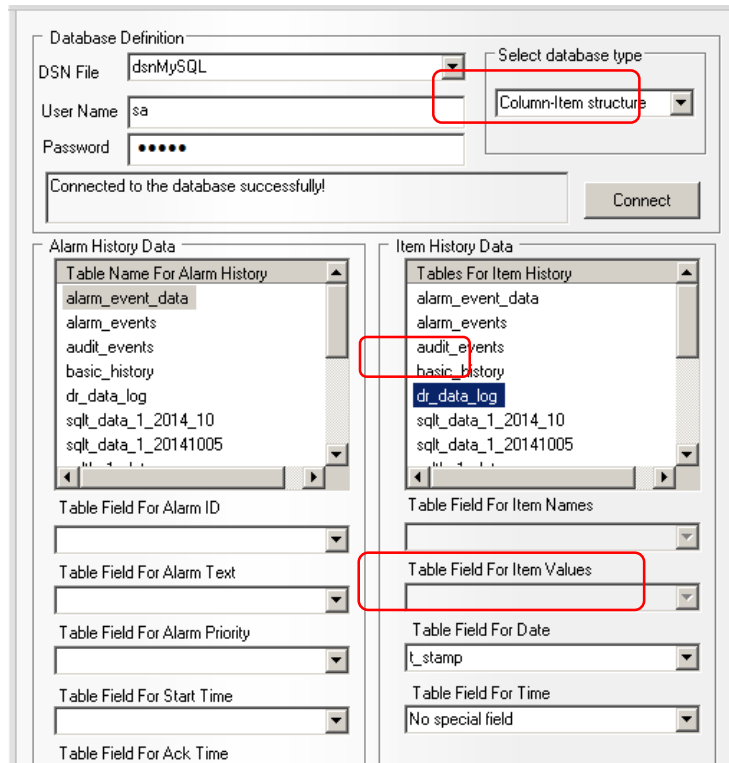


2. Enter a *Logical Name* for the driver, and then click “Configure”
3. Under the “Database Definition” section, Select the previously-created ODBC DSN from the list of DSN’s
4. Enter a valid SQL username and password, and click “Connect”
5. Once a connection is established to the database, all available tables and views will be displayed in the “Item History Data” section
6. Since the SQL view and the transaction group table have different table structures, we need to configure the ODBC driver appropriately for each connection type:
 - a. SQL View – this is a “generic” table structure that does not use the tagnames as the field names. So, we need to select the “Database Type” to be “*AnyDB structure*”, and then specify below the Table Fields that specify the Item Names, Item Values and Date:



The screenshot shows the ODBC driver configuration interface. In the 'Database Definition' section, the 'DSN File' is set to 'dsnMySQL', 'User Name' is 'sa', and the password is masked. The 'Select database type' dropdown is set to 'AnyDB structure'. Below this, the 'Item History Data' section displays a list of tables, with 'v_ignitiondata' selected. The 'Table Field For Item Names' is set to 'tagpath', 'Table Field For Item Values' is 'Value', and 'Table Field For Date' is 'DateTime'.

- b. Transaction Group Table – each of these tables are structured with each item/tagname being a column in the table. As such, we need to select the “Database Type” to be “*Column-Item structure*”, and simply select the “DateTime” field from the list for the “Table Field for Date” setting:



Database Definition

DSN File: dsnMySQL

User Name: sa

Password: ●●●●

Select database type: Column-Item structure

Connected to the database successfully!

Connect

Alarm History Data

Table Name For Alarm History

- alarm_event_data
- alarm_events
- audit_events
- basic_history
- dr_data_log
- sqlt_data_1_2014_10
- sqlt_data_1_20141005

Table Field For Alarm ID

Table Field For Alarm Text

Table Field For Alarm Priority

Table Field For Start Time

Table Field For Ack Time

Item History Data

Tables For Item History

- alarm_event_data
- alarm_events
- audit_events
- basic_history
- dr_data_log
- sqlt_data_1_2014_10
- sqlt_data_1_20141005

Table Field For Item Names

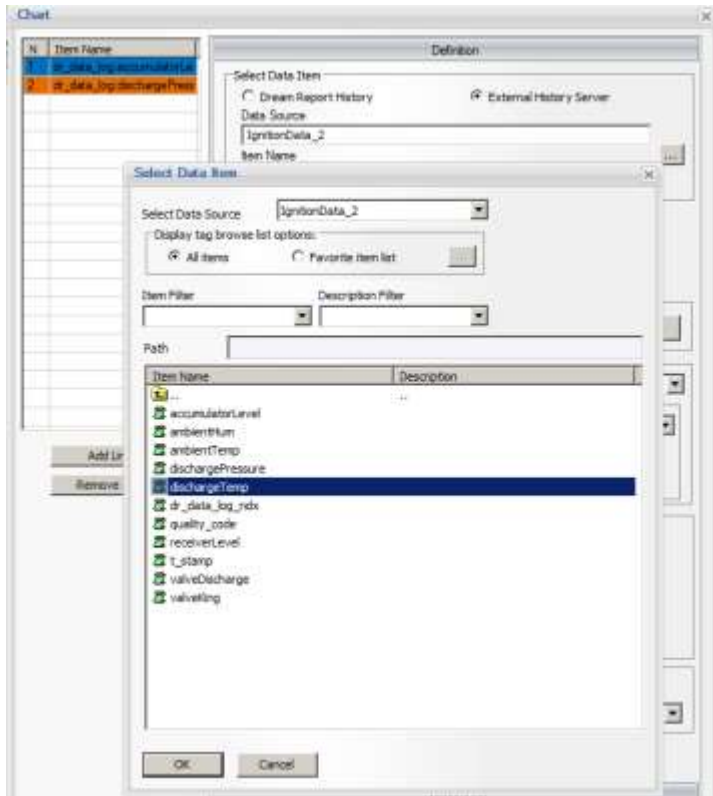
Table Field For Item Values

Table Field For Date

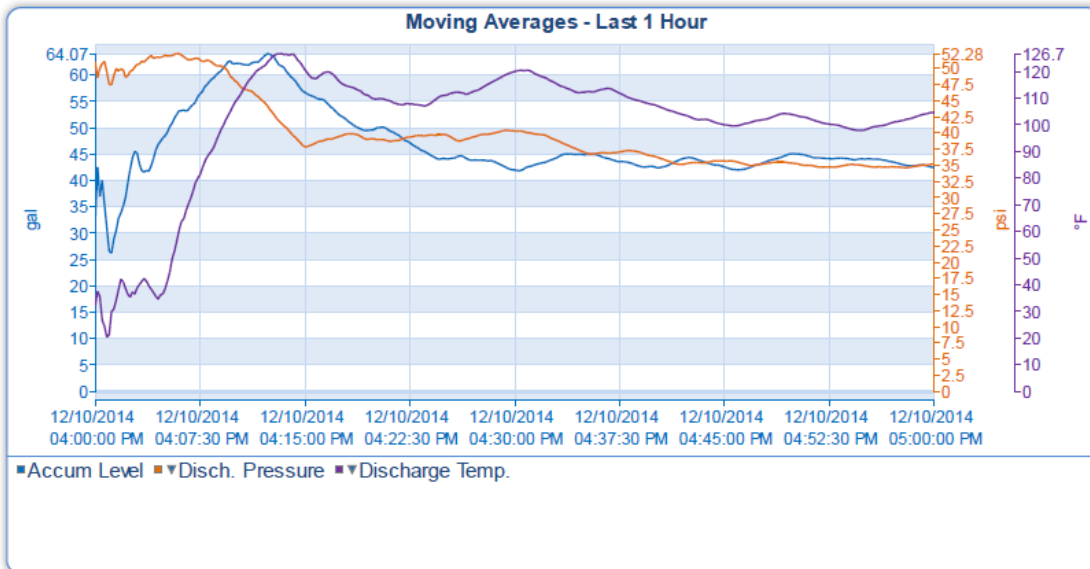
Table Field For Time

7. Once the driver has been configured, click :OK”, and then “Add”, to add it to the “Defined Drivers List”
8. Click “OK” to close the “Communication Configuration Wizard”

At this point, you are ready to start building reports. Create a new report, select any reporting object (Table, Chart, Expression, etc.), and simply browse for the tag(s) you want to add to that object. For example, to add Ignition tags to a Dream Report Line Chart object, select the “External History Server” option from the “Select Data Item” section, select the driver logical name created earlier, and a list of logged Ignition tags will be displayed:



And the resulting report:



Tagname	Max	Min	Avg	Std. Dev
Accum. Level	100.00	0.00	42.46	29.36
Disch. Pressure	60.00	2.00	35.18	17.63
Disch. Temp	212.00	0.00	104.84	69.34