# Conflict-free Replicated Data Type

(This is work-in-progress)

Currently, the following data types are supported.

1. Multi-Valued Register
2. Grow-Only Counter
3. PN-Counter
4. Enable-Wins Flag
5. Disable-Wins Flag
6. Add-Wins Observe-Remove Set
7. Remove-Wins Observe-Remove Set
8. Put-Wins Observe-Remove Map
9. Remove-Wins Observe-Remove Map
10. Recursive Composite Map

## Manual Testing of library modules for data type

Compile all the modules and add them to path.

Examples:

### Testing PN-Counter

```
% load record definition into shell
1>  rr(ec_dvv).

% create test data for PN-Counter
2> PND1 = ec_gen_crdt_test:data_pncounter(pnd1).

% run test and capture test result
3> PNR1 = ec_gen_crdt_test:run_test0(PND1).

% make query on test result
4> ec_gen_crdt_test:query_test0(PNR1).

% you should the following result
{163, 163, 163}
```

### Test Recursive Composite Map

```
% create test data for Recursive Composite Map
5> CMD2 = ec_gen_crdt_test:data_compmap02().

% run test and capture test result
 6> CMR2 = ec_gen_crdt_test:run_test0(CMD2).

% make query on test result for Put-Wins Observe-Remove Map with name pws2
```

```
7> ec_gen_crdt_test:query_test0(CMR2, [{ec_compmap, cm01}, {ec_pwormap, pws2}]).

% you should see the following result
{[{k12, [v255]}, {k11, [v254]}],
 [{k12, [v255]}, {k11, [v254]}],
 [{k12, [v255]}, {k11, [v254]}]}
```

Here you are seeing a map (**maps:to_list**) and you can further narrow your query for a key in a map given as follows:

```
8> ec_gen_crdt_test:query_test0(CMR2, [{ec_compmap, cm01}, {ec_pwormap, pws2}, k11]).

% you should see the following result
{[v254], [v254], [v254]}
```