

Overview

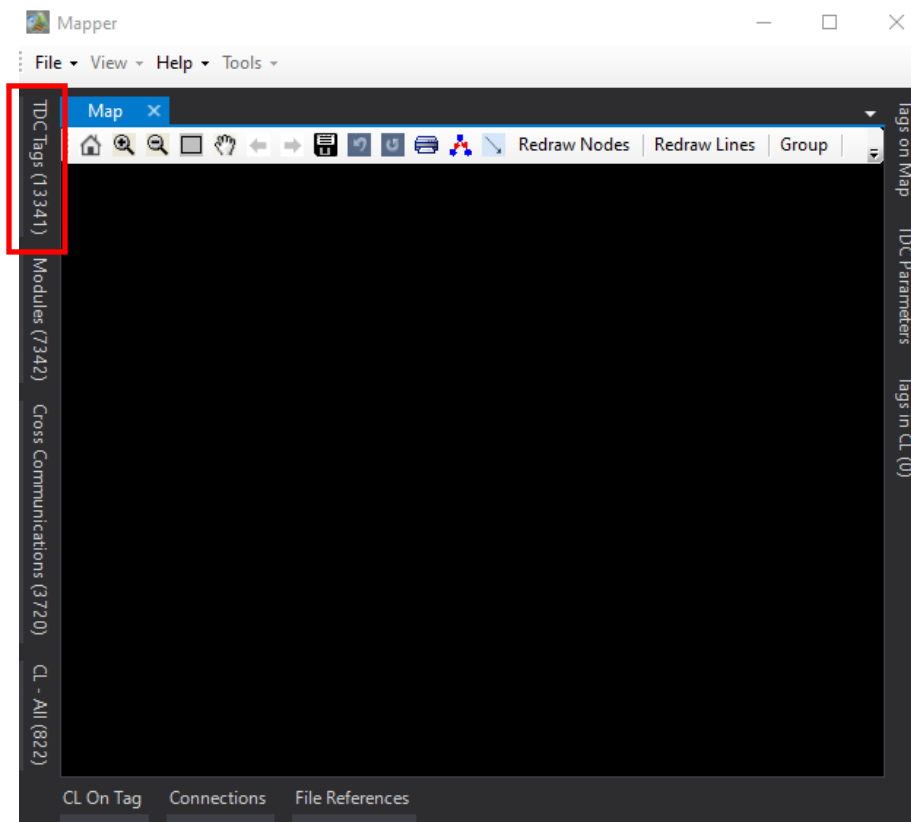
Mapper is a program that is similar to D3K. Where D3K's intended purpose is to maintain TDC systems, Mapper's intended purpose is to migrate TDC systems. Mapper shows point to point connections, maps logic blocks, maps JS files (TDC 2000), displays CL in an IDE format, formats TDC2000/3000 CL to show points that are not immediately obvious (e.g. add tagnames to \$REGCTL CDS, packages, indirect hardware addresses), and much more.

The TDC Parser (located at 10.213.10.134) is used to generate the database for mapper. Copy all the xxx.DB files into the same folder of Mapper.exe.

Mapper Windows

The mapper program is segregated into many smaller windows. These windows are hidden, unless pinned, on the edges of the app (top, bottom, left, right) and appear as tabs.

TDC Tags



TDC Tags (13341)					
Search: (All Columns)		Value for Search			
Name	PointType	Network	PVALGID	CTLALGID	ALGIDDAC
AI5625B	REGPVNIM	UCN7\HPM15	CALCULTR		
AI5625C	REGAM	AM10	GENLIN	NULL	

The “TDC Tags” window is where you can search for tags that were parsed from the EB files. When you click on a row in the table, it will pull up a diagram under the “Map” tab which displays all the interconnected nodes attached to this point.

There are several columns that you can filter on in this table which can be seen in the above screenshot.

- **Name:** this is the tag name typically defined in the EB files. There are several exceptions listed below about “BOX (Internal)” and “LOGICHG”.
- **PointType:** this is the type of tag defined in the EB files
- **Network:** this is the physical location the tag resides
- **PVALGID:** this is the parameter that defines the PV algorithm for TDC3000 points (REGPVNIM and REGAM)

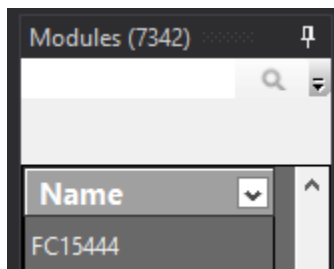
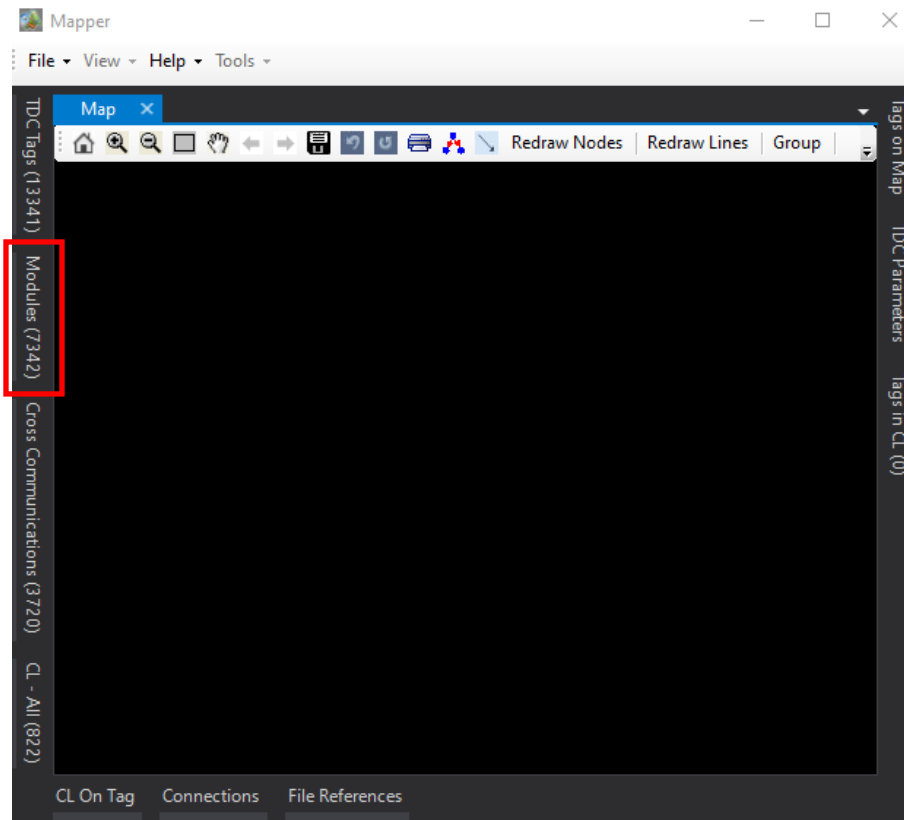
- **CTLALGID**: this is the parameter that defines the PV algorithm for TDC3000 points (REGCLNIM and REGAM)
- **ALGIDDAC**: this is the parameter that defines the PV/OP algorithm for TDC2000 points (REGHG)

There are 2 "PointType" that can be added to this table that aren't real TDC point types defined in EB files. Those are "BOX (Internal)" and "LOGICHG".

A BOX point can be thought of as a container that holds potentially hundreds of tags (e.g. FLAGS, NUMERICS, TIMERS, etc). Tags in mapper that are listed as "BOX (Internal)" are references that are used in CL or logic connections. A "BOX (Internal)" tag is only created if the parser finds that box reference being used in logic or CL.s

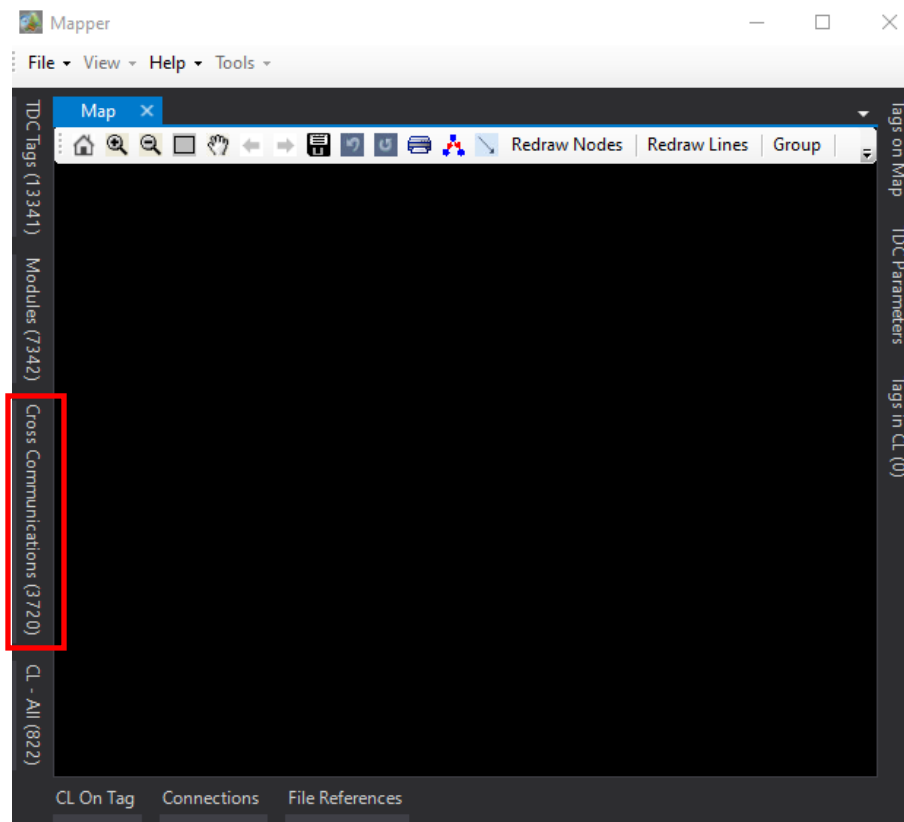
In TDC2000, you could not define logic blocks in EB files but could create logic blocks in JS files. A "LOGICHG" tag is created when there are JS files that have been parsed. These are similar to LOGICNIM blocks but are the HG equivalent.

Modules



This is a list of all the TDC Tag containers. See **Modules** for more information on how to create/delete/modify. The TDC parser automatically creates many modules for you, but they are not perfect and will need to be looked at.

Cross Communications



Cross Communications (3720)			
(All Columns)		Value for Search	
Source Tag	Target Tag	Source Network	Target Network
FI4219	FY4116A	HWY1\HLPIU8	AM10
FI4467S	FD4462	UCN6\SM15	AM10

This is a useful table if the entire LCN is not being migrated, in particular, it's good for quickly identifying AM tags that should and shouldn't be migrated. The table lists all the parameter connections (i.e. not CL connections) between 2 tags that occur on different networks. Clicking on a row will pull up a map of the tags.

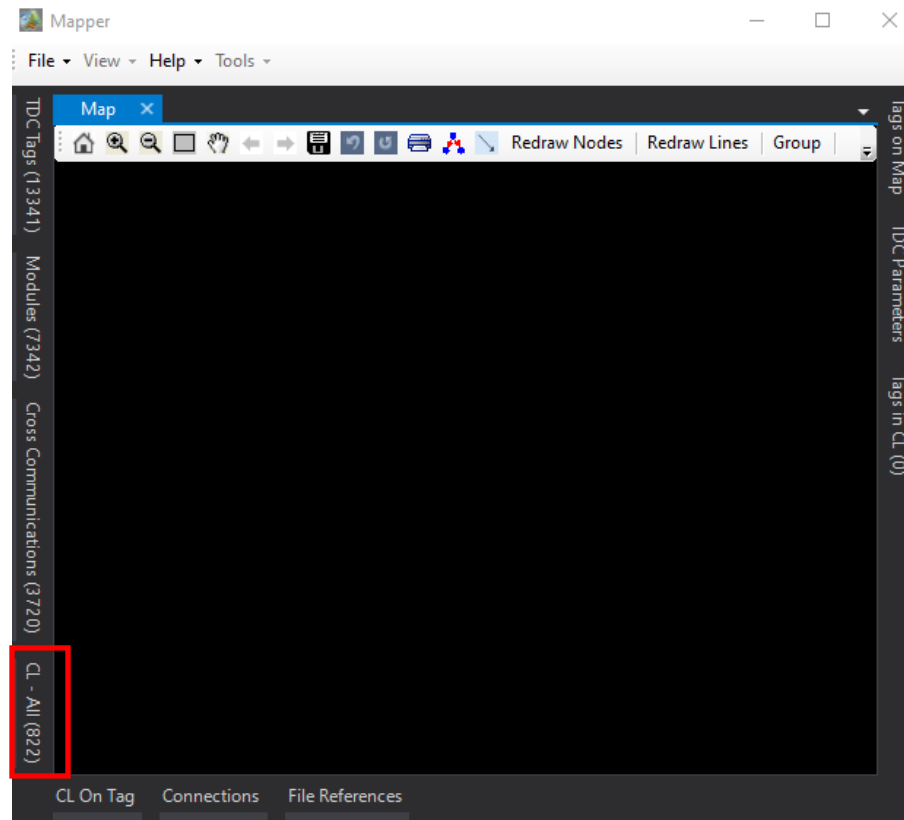
Consider the following scenario based on the screenshot above.

- All of UCN6 is being migrated,
- AM points are being migrated on a case-by-case basis.

- HWY1 is not being migrated

You can quickly filter “in scope” tags based off of “Source Network”. From that list, you could further filter “Target Network” to see all AM tags. Assuming there are no other connections to these tags or CL involved, you can quickly see that “Target Tag” FD4462 should probably be “in scope” and FY4116A should probably not be in scope

CL - All



CL - All (822)			
Search:	(All Columns)	Value for Search	
FileName	TagName	PointType	Network
MAX_MIN4.CL (Package)	XAI5626	REGAM	AM10
MAX_MIN4.CL (Package)	XAI5825	REGAM	AM10
MAX_MIN4.CL (Package)	XAI5826	REGAM	AM10
MAX_MIN4.CL (Package)	XAI5960	REGAM	AM10
MAX_MIN4.CL (Package)	XAI5961	REGAM	AM10
MAX_MIN4.CL (Package)	XTI5534A	REGAM	AM10
MAX_MIN4.CL (Block)	XTI5734A	REGAM	AM10
M_AI.CL	ZI26095A	REGAM	AM41
M_AI1.CL			

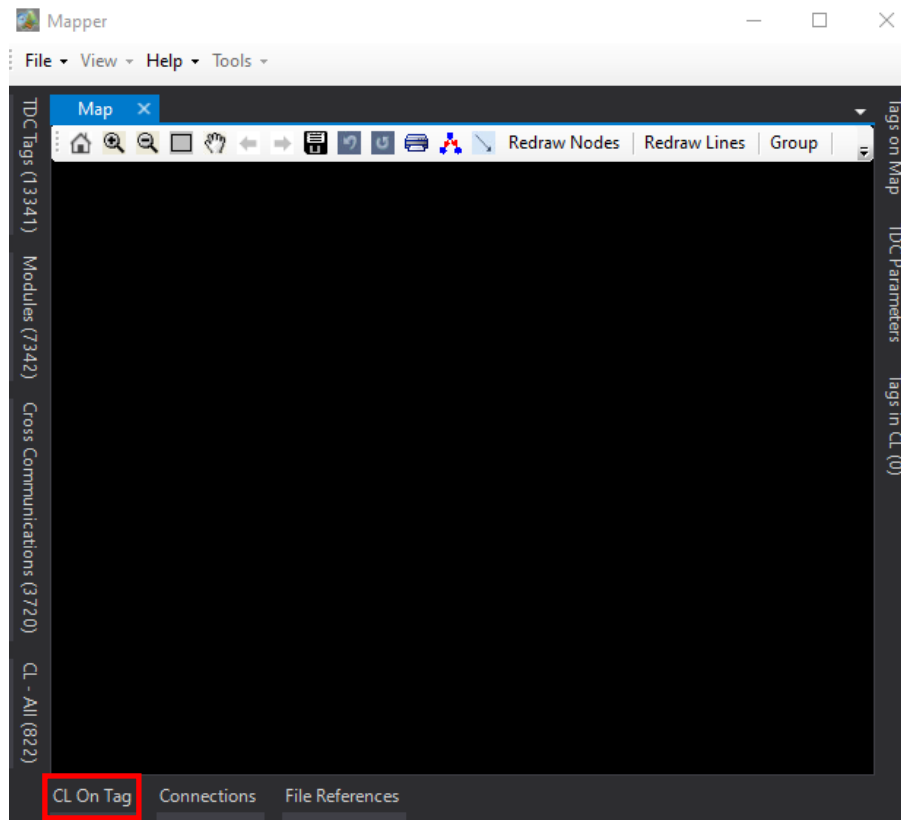
This window displays all the parsed CL files. Clicking on a cell will pull up a new tab or jump to an existing tab that displays the CL in a formatted manner. This table allows you to quickly find “in scope” and “out of scope” CL by filtering on “Network”. The columns are defined as follows:

- **FileName**: this is the name of the CL file
- **TagName**: the CL is linked to this tag i.e. the CL program runs inside of this tag
- **PointType**: the type of tag that the CL is linked to
- **Network**: the physical location of the tag that the CL is linked to

There is a row in this table for every CL file. In addition to this, if a CL program is a PACKAGE, each tag the CL file links to is also listed in this table. The parser searches for parameters in the EB file named PKGNAME(i). The value of the PKGNAME parameter is a CL file. This allows the parser to figure out how CL is linked to TDC tags when they are not explicitly defined. You will see “(Package)” appended to the end of “FileName” if the PKGNAME parameter is found on a tag. Sometimes the PKGNAME parameter is not defined in the EB exports for some reason (perhaps a bad export, but I’m not sure). In this situation, if the CLSLOTS.ZB file was defined during parsing, the parser will check this file to see how tags are linked to CL. If the parser discovers a link in this file and not on the PKGNAME parameter, then you will see “(Block)” appended to the end of “FileName”.

CL files that have no tag defined likely do not need to be explicitly migrated. These are often parameter lists or unlinked (broken) CL.

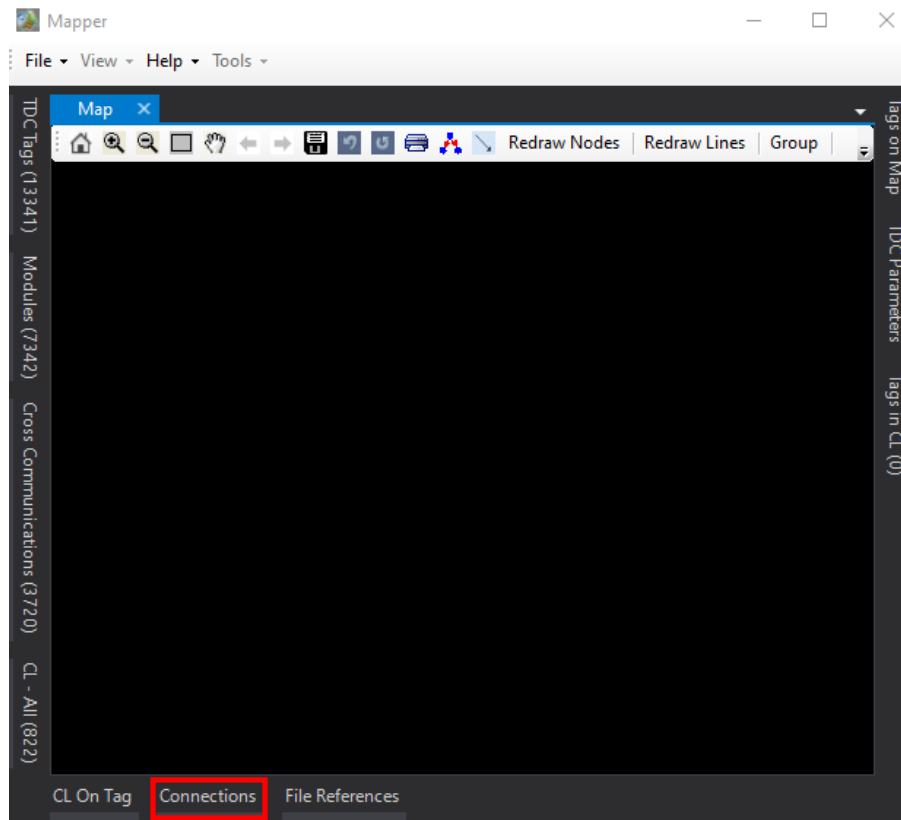
CL On Tag



CL On Tag (2)	
Search:	(All Columns)
Name	
(FI24010G)	
EDC_DENS.CL	
LEVEL.CL	

This table displays the CL that references the selected tag. The selected tag is shown in parenthesis and appended to the end of "Name" column e.g. (FI24010G) is selected tag in screenshot above. Whenever your selected tag changes (click a tag on the "Map", click a tagname in a CL file, etc) then this table will update. Clicking on a cell in will pull up the CL file.

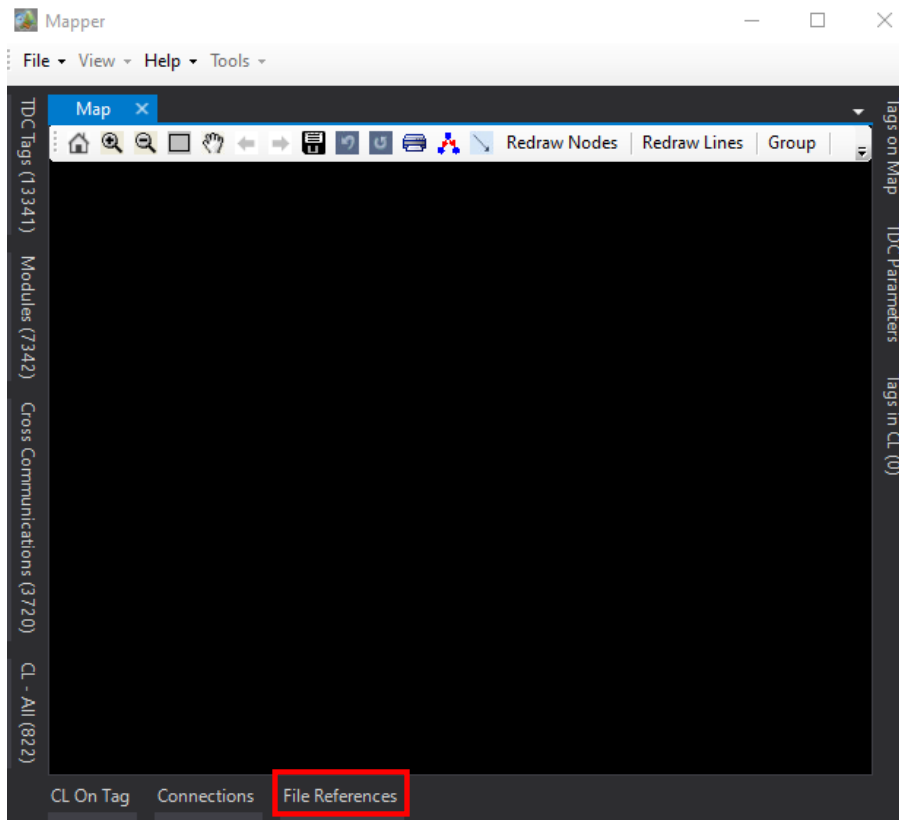
Connections



Connections (7)	
Search: (All Columns) Va	
Connection	Parameter
PC9020A.PISRC(1)	SP

This table displays the parameter connections that references the selected tag. Whenever your selected tag changes (click a tag on the “Map”, click a tagname in a CL file, etc) then this table will update. The “Connection” column is connected to the “Parameter” column which can be visually seen via the “Map”.

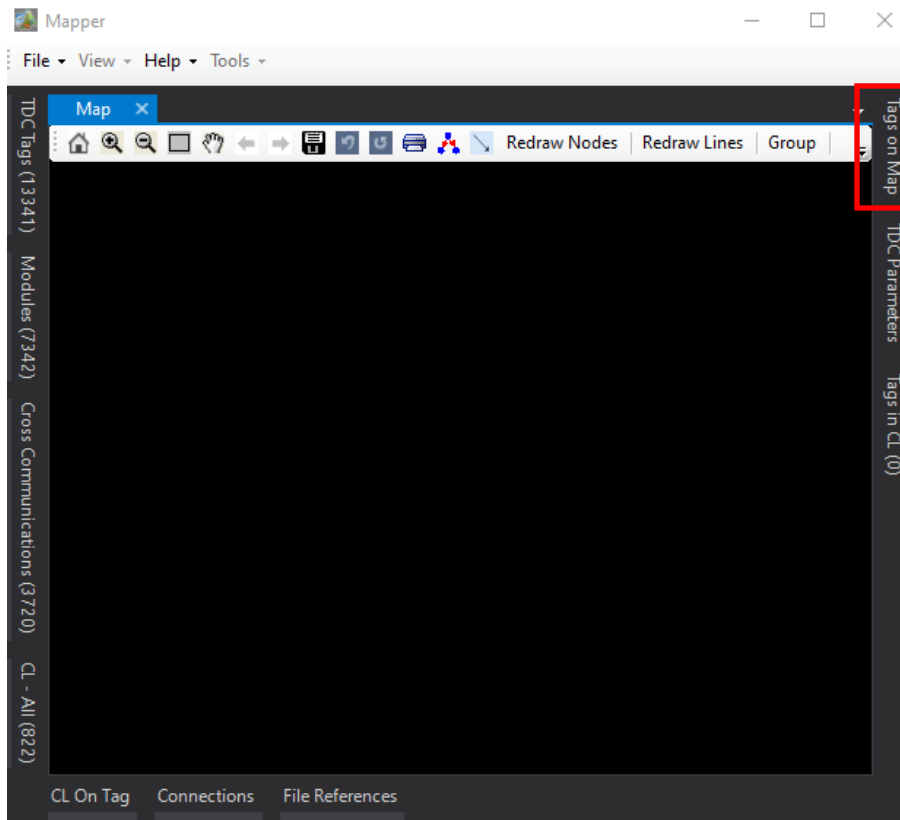
File References



File References (3)	
Search:	(All Columns) Value for Search
FileType	Value
FFL	ENPRDLOG
Group	1:171
Schematic	D:NET>USF1>GR01>501.DS

This table displays miscellaneous files that reference the selected tag. Whenever your selected tag changes (click a tag on the “Map”, click a tagname in a CL file, etc) then this table will update. This data is only available if a D3K.mdb file was included during parsing. The table displays all the Free format log (FFL), Group, and Schematic (graphic) references.

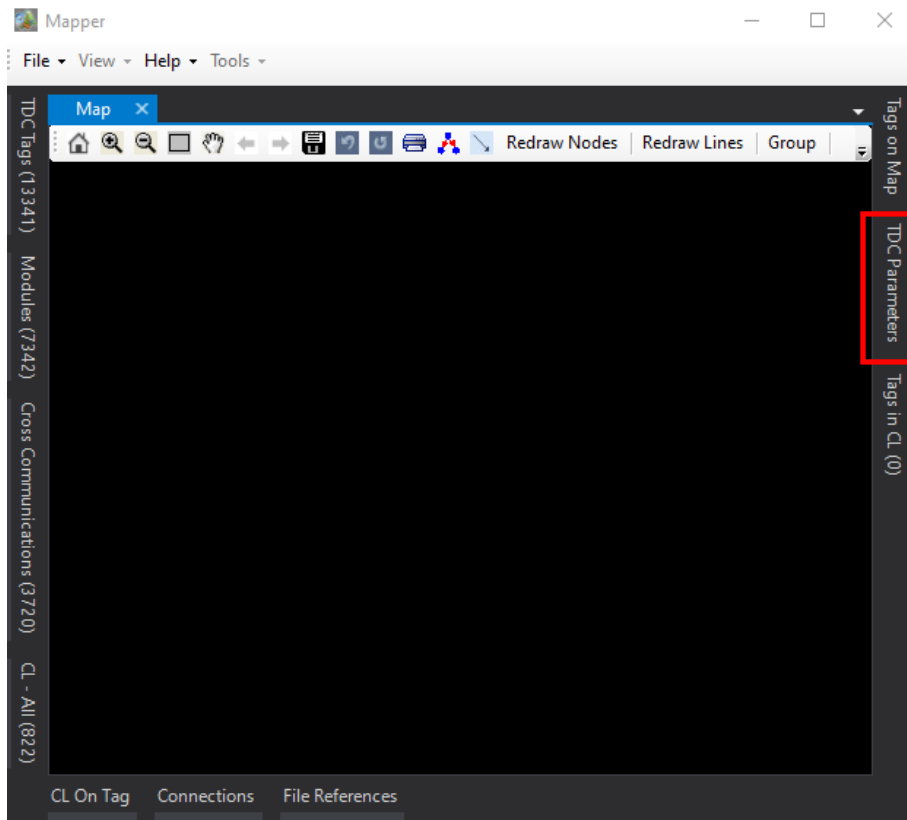
Tags On Map



Tags on Map (10)		
(All Columns)	Value for Search	
Name	PointType	Address
AI5825	ANINNIM	UCN7\HPM15

This table shows all the tags that are in the current “Map”. Clicking on a cell zooms into the node of the “Map”. Use the scroll wheel to zoom back out to a reasonable view from there.

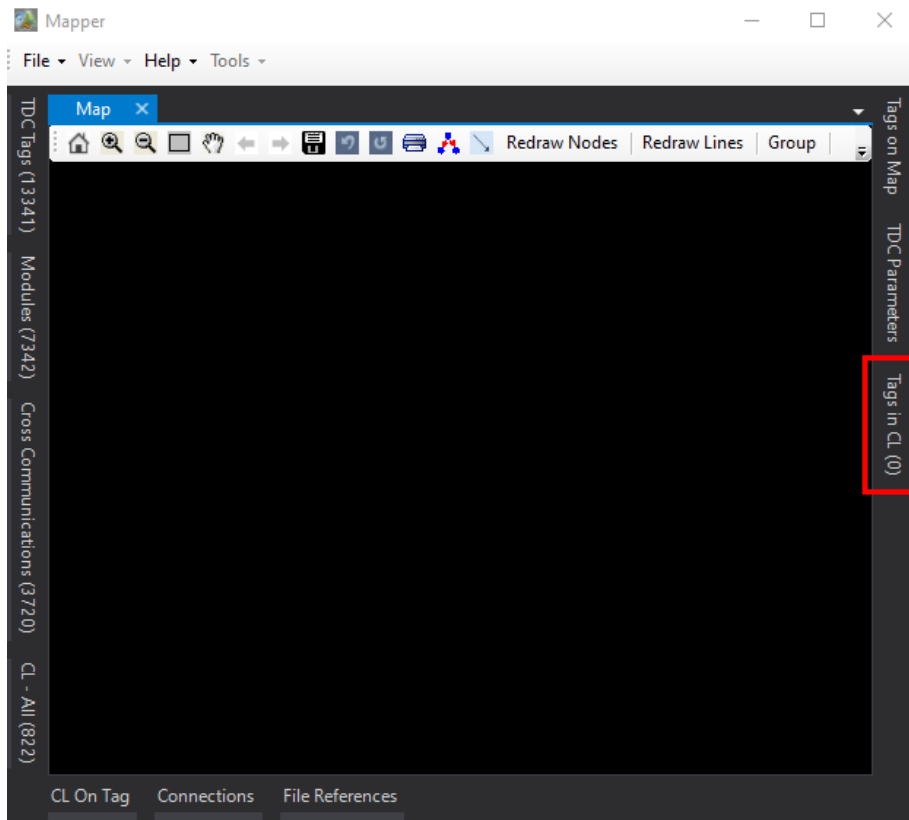
TDC Parameters



TDC Parameters (79)	
(All Column)	Value for Search
Name (XAI5825)	Value
PointType	REGAM
Network	AM10
SAUXUNIT	
SDHSELECT	CUTOUT

This table displays all the parameters defined in the EB files of the selected tag. Whenever your selected tag changes (click a tag on the “Map”, click a tagname in a CL file, etc) then this table will update.

Tags In CL



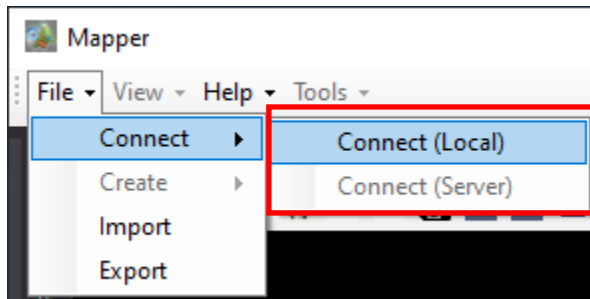
Tags in CL (37)		
(All Columns) Value for Search		
Name	PointType	Network
100_USE	REGAM	AM10
FC4463	REGCLNIM	UCN6\HPM13
FC5100	REGHG	HWY2\MC20

This table shows all the tags that are defined in the open CL file.

File

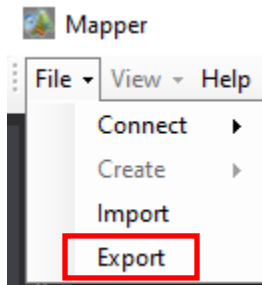
There are several options under “File” in the toolbar such as “Connect”, “Import”, “Export”

Connect

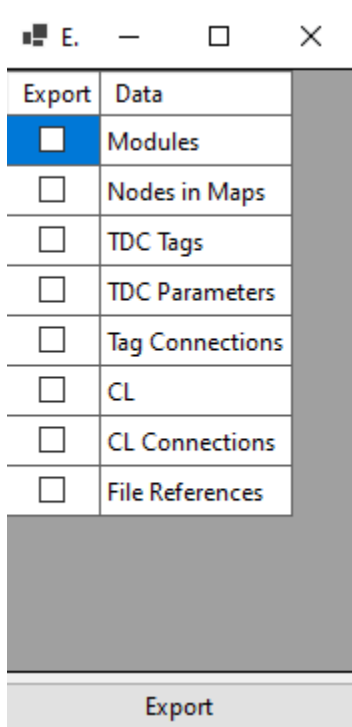


You can have several DB files stored in different locations that Mapper can open.

Export



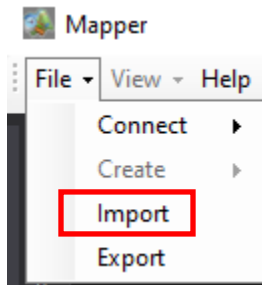
You can connect any application that is capable of opening a SQLITE DB file to the “.DB” file of mapper. However, you can also use the “Export” button to easily export the data to excel. Checking multiple boxes will add sheets to the spreadsheet.

A screenshot of the 'Export' dialog box in the Mapper application. The dialog has a title bar with 'E.' and standard window controls. It contains a table with two columns: 'Export' (checkboxes) and 'Data' (labels). The 'Export' column has a blue header, and the 'Data' column has a white header. The table lists several data types with checkboxes next to them. At the bottom of the dialog is an 'Export' button.

Export	Data
<input checked="" type="checkbox"/>	Modules
<input type="checkbox"/>	Nodes in Maps
<input type="checkbox"/>	TDC Tags
<input type="checkbox"/>	TDC Parameters
<input type="checkbox"/>	Tag Connections
<input type="checkbox"/>	CL
<input type="checkbox"/>	CL Connections
<input type="checkbox"/>	File References

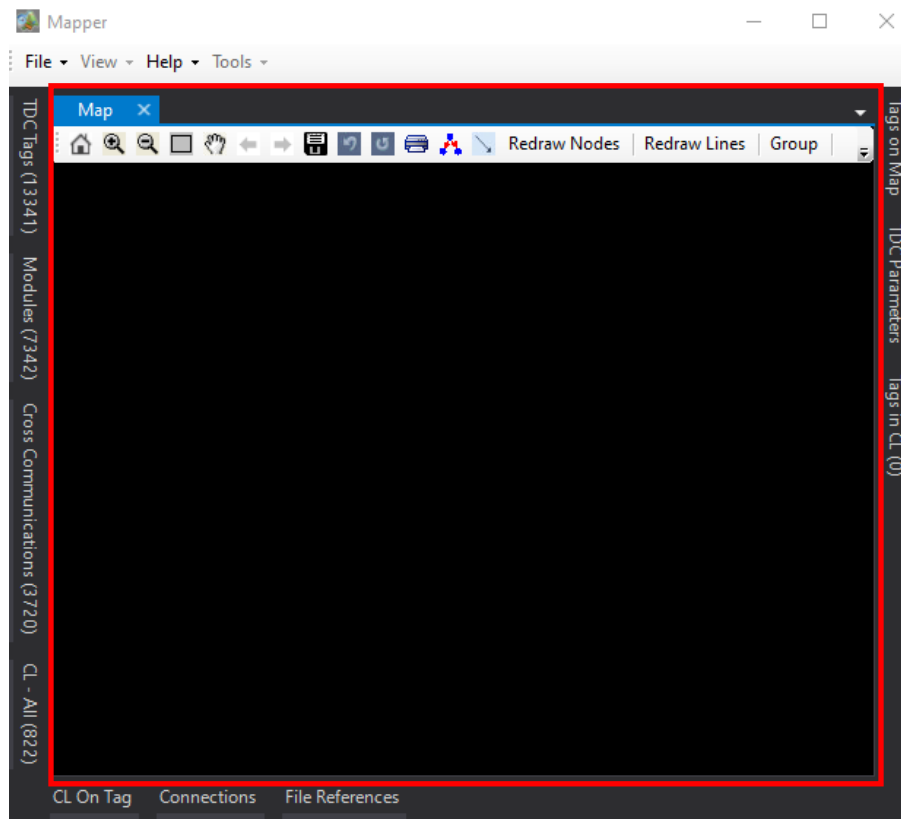
Export

Import

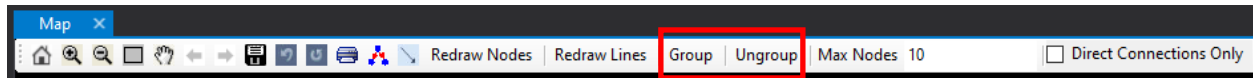


You can bulk edit changes from an exported file and then re-import the spreadsheet. There is a column called "Delete" that you can set to true/false if you want to remove something. If you want to add a row, just fill in all the necessary columns. This is useful to bulk edit tags in modules.

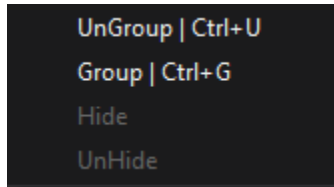
TDC Map



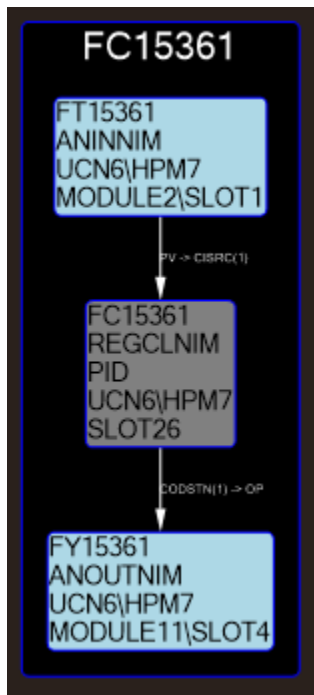
Modules



The TDC parser automatically creates modules and assigns tags to these modules. However, you may need to tweak these modules. You use the “Group” and “Ungroup” button to do this. You can also use the context menu by right clicking anywhere on the map or the shortcuts listed on the context menu.



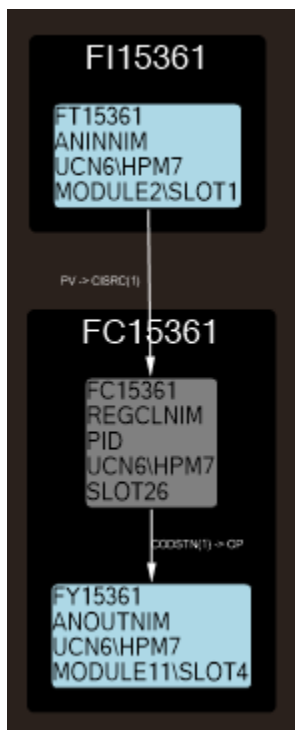
Select the tags that you want to ungroup (blue outline indicates tag is selected) then press “Ungroup”



The result looks like this

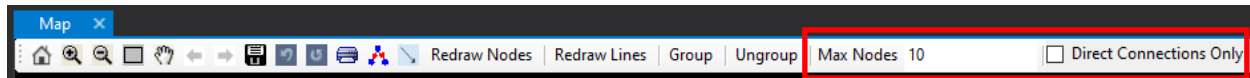


You can then select tags however you want and click “Group”, fill in the name you’d like for the module then click “Ok”. You could for instance, change the module to this



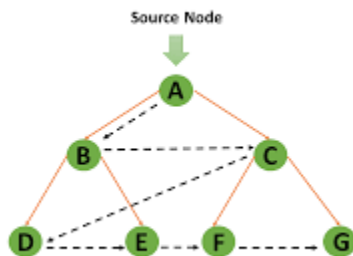
Out of all the selected tags, if 1 of them has a defined module, all the tags will be merged into that same module. If none of the tags have a module or if there are tags in multiple modules, then you will be prompted for a new module name.

Reducing Clutter



The “Max Nodes” textbox and “Direct Connections Only” checkbox control how many nodes are displayed on the map. Often times you don’t need to see the entire map because it just makes it more difficult to digest the information you can see.

Changing the value in “Max Nodes” will start from the selected tag and then do a breadth width first (BFW) search to find the closest nodes. See below screenshot for example.



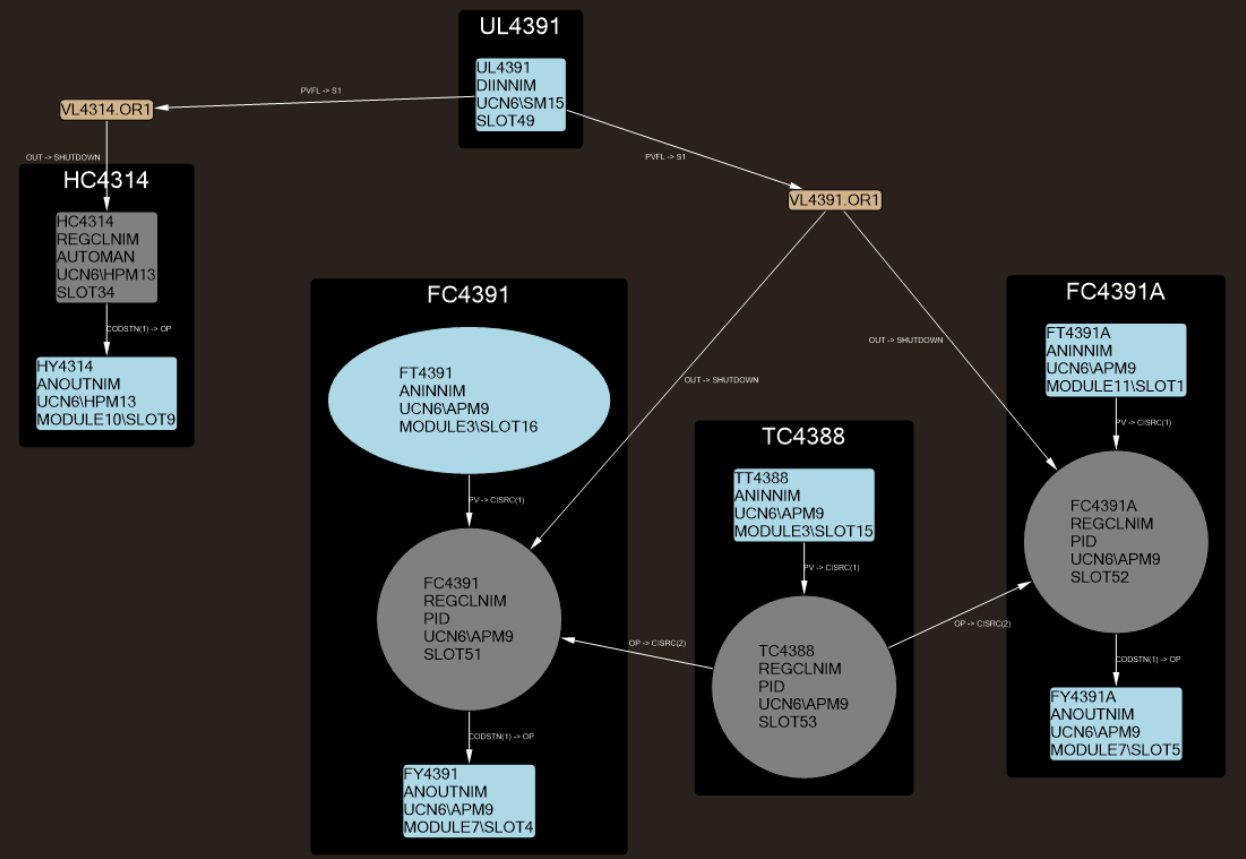
“Direct Connections Only” will only show tags that are directly connected to the selected tag via parameter connections.

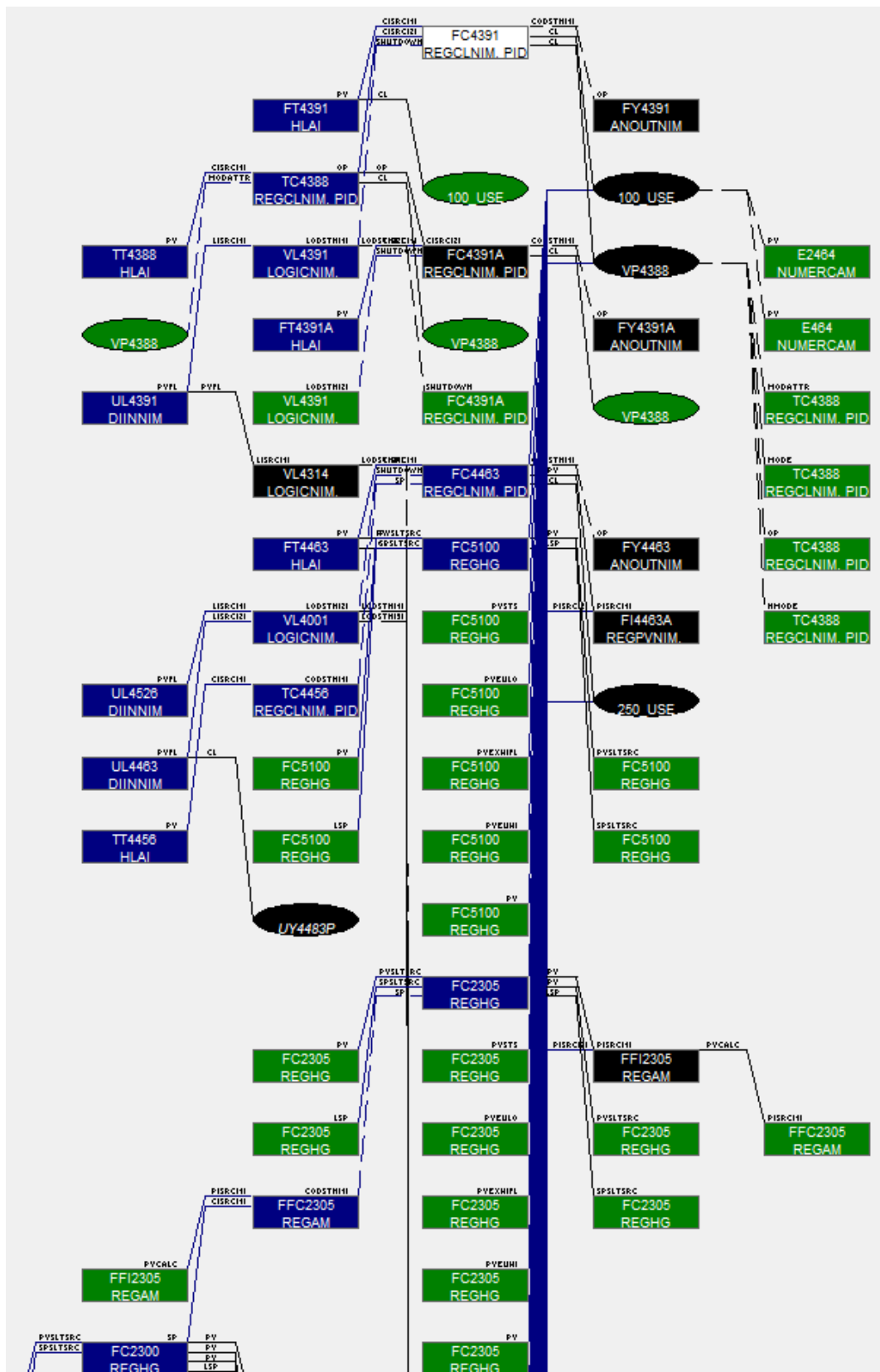
Tags that are hiding connections have a red outline.



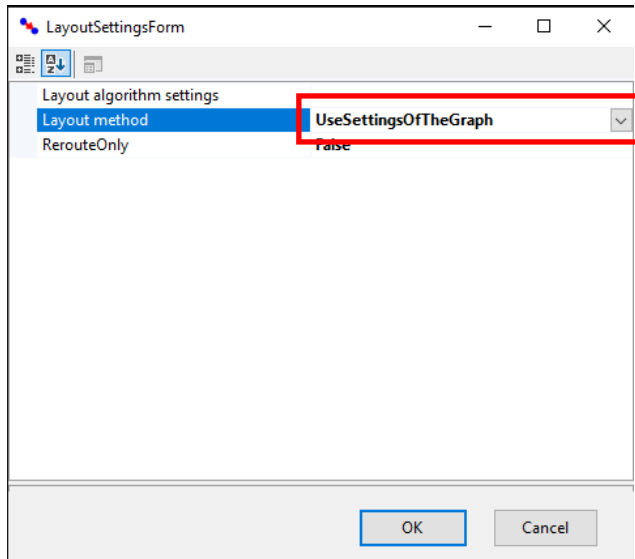
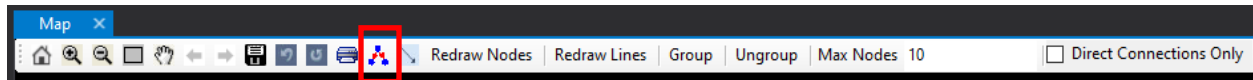
There are several key differences in how “Mapper” creates a diagram of interconnected nodes compared to DOC3000. The main reason for the discrepancies is a “Map” can become saturated with tags that often don’t affect one another. Below the discrepancies you will find 2 screenshots that are mapping the same tag.

- Mapper only displays a tag once whereas D3K could display the same tag numerous times. See D3K screenshot below, all green tags are duplicates.
- The “LOGICNIM” can be thought of as a container for many more tags. Mapper automatically breaks apart “LOGICNIM” tags (“Map Logic Block” in D3K) into separate logic gates i.e. you will never see a “LOGICNIM” tag on a map, however, you will see the logic gate on the map (see tan tags on Mapper screenshot).
- CL isn’t displayed as nodes on a map in Mapper but is displayed in D3K. D3K displays CL files with oval nodes in a map. Mapper indicates a tag has CL on it by using an oval shape. It’s a small difference but the outcome is a much less cluttered map



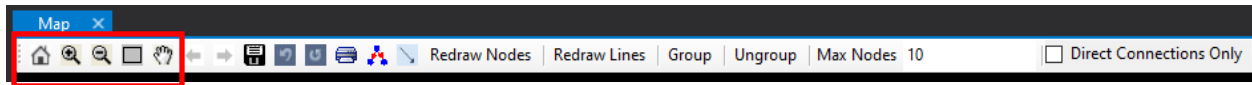


Drawing settings



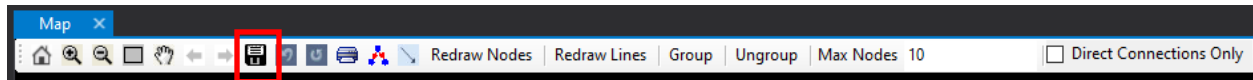
You can change the algorithm for how a map is displayed on a map. You would likely only need to do this for very large maps and you are probably better off simply reducing the “Max Nodes”

Pan & Zoom



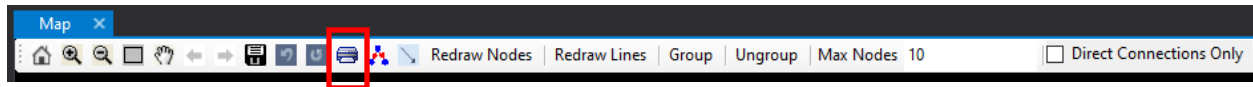
You can use the selected buttons to reset zoom, zoom in, zoom out, zoom to selection, and pan. Alternatively, you can scroll up to zoom in, scroll down to zoom out, and click and hold scroll wheel to pan.

Save Map



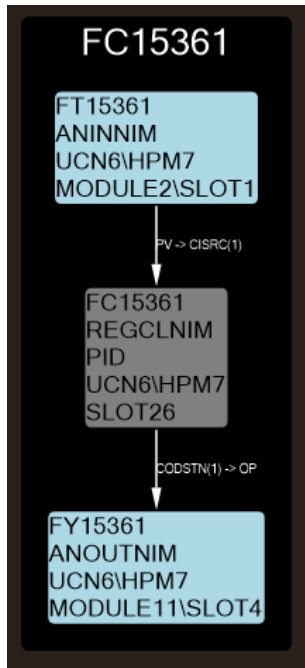
This feature does not work at this time. Plans to save node location if user wants to manually manipulate map.

Print Map



Prints the map to PDF or printer.

Function Block Colors



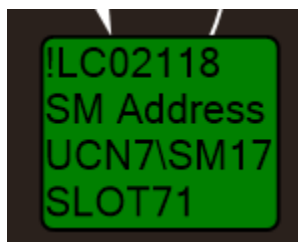
- Black container is module
- Blue is IO
- Gray is point type that isn't IO



- Red outline is missing connections



- Tan is not real TDC tag as defined by EB files. It is auxiliary logic.



- Green is SM address



- Red is missing tag

Function Block Shapes



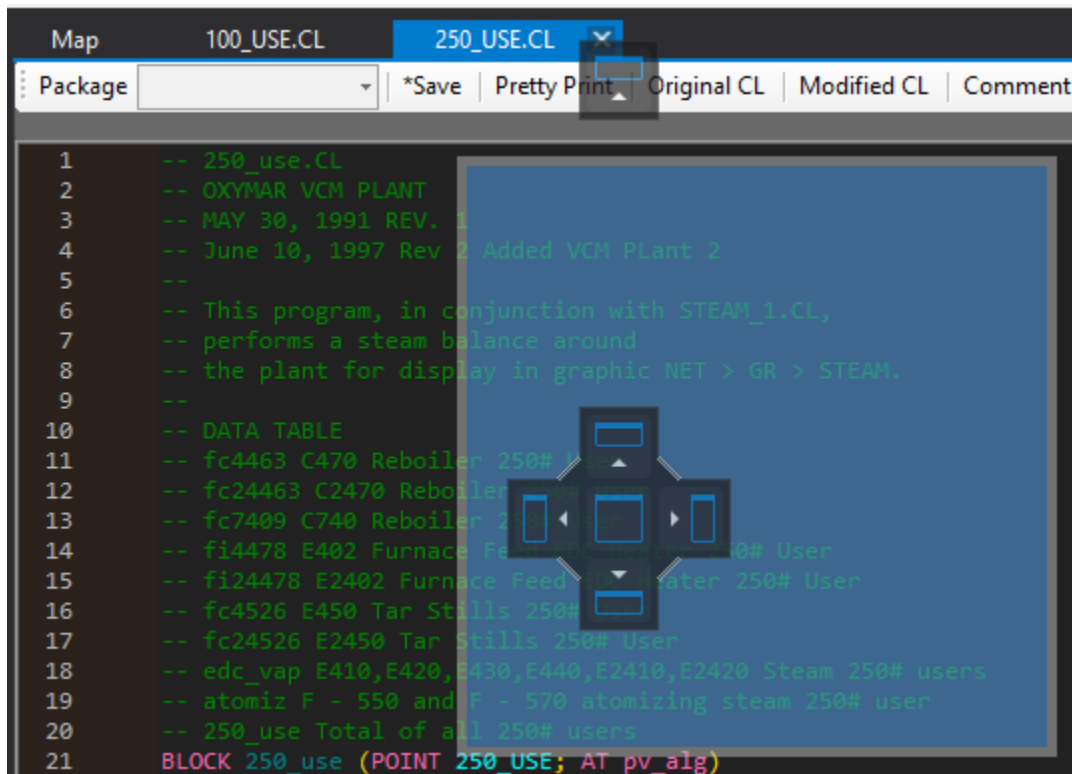
- Oval indicates tag appears in CL
- Rectangle is no CL on tag

CL

See “Help” toolbar in Mapper, then click CL for an interactive guide of the most important elements of the program.

Multiple CL

You can open as many CL as you want and they will be added to the top tab. You can also drag tabs so that you have 2 CL files side by side.



The screenshot shows the CL editor interface. At the top, there are two tabs: "100_USE.CL" and "250_USE.CL". Below the tabs is a toolbar with buttons for "Package", "Save", "Pretty Print", "Original CL", "Modified CL", and "Comment". The main area is divided into two panes. The left pane contains a code editor with the following text:

```
1  -- 250_use.CL
2  -- OXYMAR VCM PLANT
3  -- MAY 30, 1991 REV. 1
4  -- June 10, 1997 Rev 2 Added VCM PLant 2
5  --
6  -- This program, in conjunction with STEAM_1.CL,
7  -- performs a steam balance around
8  -- the plant for display in graphic NET > GR > STEAM.
9  --
10 -- DATA TABLE
11 -- fc4463 C470 Reboiler 250#
12 -- fc24463 C2470 Reboiler
13 -- fc7409 C740 Reboiler
14 -- fi4478 E402 Furnace Feed 250# User
15 -- fi24478 E2402 Furnace Feed 250# User
16 -- fc4526 E450 Tar Stills 250#
17 -- fc24526 E2450 Tar Stills 250# User
18 -- edc_vap E410,E420,E430,E440,E2410,E2420 Steam 250# users
19 -- atomiz F - 550 and F - 570 atomizing steam 250# user
20 -- 250_use Total of all 250# users
21 BLOCK 250_use (POINT 250_USE; AT pv_alg)
```

The right pane contains a diagram editor with a light blue background. It shows a process flow diagram with several rectangular blocks connected by arrows. The blocks are labeled with the same text as the code editor: "fc4463 C470 Reboiler 250#", "fc24463 C2470 Reboiler", "fc7409 C740 Reboiler", "fi4478 E402 Furnace Feed 250# User", "fi24478 E2402 Furnace Feed 250# User", "fc4526 E450 Tar Stills 250#", "fc24526 E2450 Tar Stills 250# User", "edc_vap E410,E420,E430,E440,E2410,E2420 Steam 250# users", "atomiz F - 550 and F - 570 atomizing steam 250# user", and "250_use Total of all 250# users". The diagram shows the flow of material between these units.

Mapper
File View Help Tools

Map
100_USE.CL
Package
*Save Pretty Print Original CL

250_USE.CL
Package
*Save Pretty Print Original CL

TDC Tags (1341)
Modules (7342)
Cross Communications (3720)
CL - All (822)

```

1  -- 100_use.cl AM CL Program
2  -- OXYMAR VCM PLANT
3  -- MAY 30, 1991 REV. 1
4  -- June 10, 1997 Rev 2 Added VCM Plant 2
5  --
6  -- This program, in conjunction with STEAM_1.CL,
7  -- performs a steam balance around
8  -- the plant for display in graphic NET > GR > STEAM
9  --
10 -- DATA TABLE
11 -- FC5100 C522 100# Stripping Steam
12 -- FC2305 C210 Reboiler 100# User
13 -- FC22305 C2210 Reboiler 100# User
14 -- FC2344 C220 Reboiler 100# User
15 -- FC2527 C240 Reboiler 100# User
16 -- FC4391 C460 Reboiler 100# User
17 -- FC24391 C2460 Reboiler 100# User
18 -- FC24968 C2460 Reboiler Trim 100# User
19 -- FC3294 C360 Ethylene Stripping Steam 100# User
20 -- FC23294 C2360 Ethylene Stripping Steam 100# User
21 -- FI4424 E464 HCl Vapor Flow
22 -- FI24424 E2464 HCl Vapor Flow
23 -- C230 C230 Reboiler 100# user
24 -- Y598 C522 Steam Eductor 100# user
25 -- H360 C360 Steam Eductor 100# user
26 -- Y2598 C2522 Steam Eductor 100# user
27 -- H2360 C2360 Steam Eductor 100# user
28 -- E301 E301 HCl Heater 100# user
29 -- E2301 E2301 HCl Heater 100# user
30 -- E302 E302 Ethylene Heater 100# user
31 -- E2302 E2302 Ethylene Heater 100# user
32 -- E303 E303 Oxygen Heater 100# user
33 -- E2303 E2303 Oxygen Heater 100# user
34 -- 100_use Total of all 100# users
35 -- E901 E901 Ethylene Letdown Heater 100# user
36 -- E2901 E2901 Ethylene Letdown Heater 100# user
37 BLOCK 100_use (POINT 100_USE; AT pv_alg)
38
39     EXTERNAL FC4463, FC5100, FC2305, FC2344, FC2527,
40         & FC4391, FC3294, FI4424, FT4391
41
42     EXTERNAL C230, Y598, H360,
43         & E901, E303, E301,E302
44
45     EXTERNAL FC24463, FC22305, FC24391, FC24968,
46         & FC23294, FI24424,
47         & E2464, E464, Y2598,
48         & H2360, E2301, E2302, E2303, E2901
49
50     EXTERNAL FT24463, FT22305, FT23294, FT2527,
51         & FT24391, FT24968
52
53     LOCAL 100_misc = 2.0 * 2 -- plant 2 and 1
54     LOCAL i1, i2, i3, i4
55     LOCAL c220, c210, c240, c360, c460,

```

```

1  -- 250_use.CL
2  -- OXYMAR VCM PLANT
3  -- MAY 30, 1991 REV. 1
4  -- June 10, 1997 Rev 2 Added VCM Plant 2
5  --
6  -- This program, in conjunction with STEAM_1.CL,
7  -- performs a steam balance around
8  -- the plant for display in graphic NET > GR > STEAM
9  --
10 -- DATA TABLE
11 -- FC4463 C470 Reboiler 250# User
12 -- FC24463 C2470 Reboiler 250# User
13 -- FC7409 C740 Reboiler 250# User
14 -- FI4478 E402 Furnace Feed EDC Heater 250# User
15 -- FI24478 E2402 Furnace Feed EDC Heater 250# User
16 -- FC4526 E450 Tar Stills 250# User
17 -- FC24526 E2450 Tar Stills 250# User
18 -- edc_vap E410,E420,E430,E440,E2410,E2420 Steam 250#
19 -- atomiz F = 550 and F = 570 atomizing steam 250#
20 -- 250_use Total of all 250# users
21 BLOCK 250_use (POINT 250_USE; AT pv_alg)
22
23     EXTERNAL FC4463, FC7409,
24         & FC4535, FI4478
25
26
27     EXTERNAL EDC_VAP1, EDC_VAP2, ATOMIZ
28
29     EXTERNAL FC24463, FC24535, FI24478,
30         & FI24424
31
32     EXTERNAL FT24463,
33         & FT24535
34
35     LOCAL k650_lo = 6.2
36     LOCAL k2650_lo = 6.2
37     LOCAL k350_lo = 3.8
38     LOCAL k2350_lo = 3.8
39     LOCAL scrbr_pu = 15.0
40     LOCAL i1, i2, i3, i4
41     LOCAL c470, e402, c740,
42         & tarstill,
43         & c2470, e2402,
44         & E2450
45
46     -- Check inputs and set to zero if bad
47
48     IF FC4463.pvsts = BAD THEN
49         & SET c470 = FC4463.pveulo
50     ELSE IF FC4463.pvhifl = on THEN
51         & SET c470 = FC4463.pveuhi * 1.029
52     ELSE IF FC4463.pv < FC4463.pveulo THEN
53         & SET c470 = FC4463.pveulo
54     ELSE SET c470 = FC4463.pv
55

```