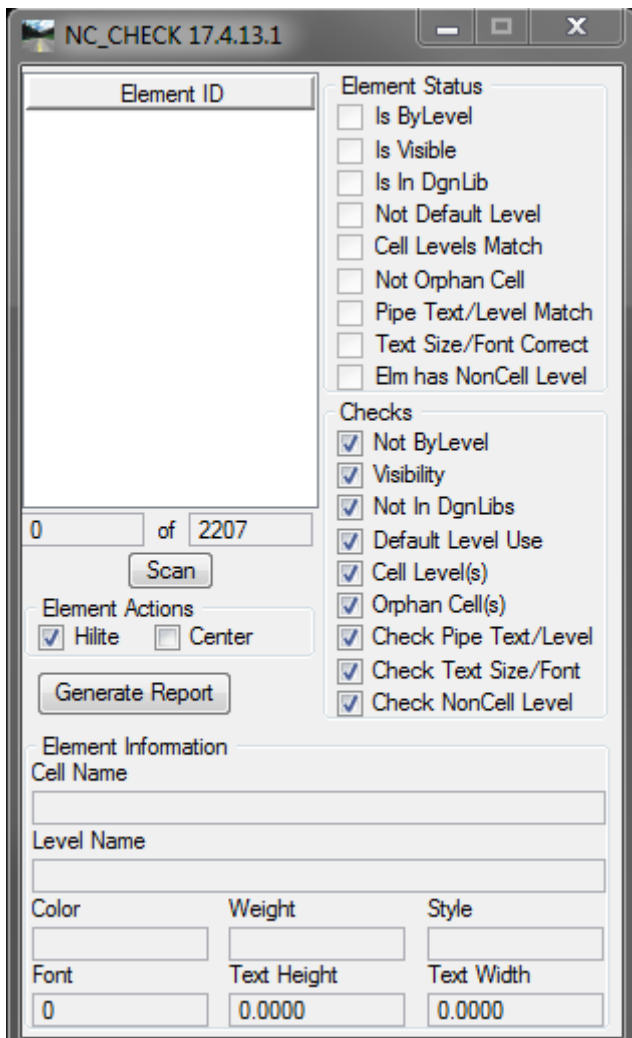


NC_Check Instructions and Information

NC_Check is an NCDOT program written by Patrik Holbrook. The main purpose of the program is to provide a quick and consistent means to ensure the elements shown in a dgn file meet the standards set forth by the Mapping Manual and the NCMMap program. The program evaluates each element in the file against several resources to make sure they match. These resources range from dgnlibs located in our Workspaces, Excel CSV files, cell libraries and comparisons within the file itself.



The NC_Check dialog is divided into 5 main areas. The Element ID window, in the upper left portion of the dialog, shows each individual element by its ID number that does not match the criteria selected in the dialog. Only one element at a time can be selected in this window and is represented by the blue highlight. Below the window is a readout showing the number of element outliers there were for this check, a total number of elements in the Active dgn/model and a Scan button to start the program scanning the file for the selected characteristics.

The next one is the Element Status area, on the upper right side. This list shows the particular characteristic of the selected element and does not accept user input.

Below it is the Checks area. The user should check on the different characteristics they wish to check the elements for. These checks are:

Not ByLevel – All element placed into a dgn file utilizing NCMMap, Geopak or SMD procedures should have the ByLevel attribute assigned to them.

Visibility – This will reveal (Hi-lite on) any lines that have been generated but are not normally visible in the drawing. The Visibility attribute is totally separate of the Transparency attribute. The NCDOT Utility Counter Program previously would count the “invisible” lines according to its attributes. The current NCDOT Utility Counter Program, does not count these “invisible” lines, but as a “best practice,” these lines should be eliminated. A

possible cause for the visibility issue seems to be "Gap Lines," [Lines that are created due to inadequate Begin Line / End Line field codes.]

Not In Dgnlibs – This checks to make sure that all used levels in the Active file are in one of the configured dgnlibs. The list of dgnlibs checked against are defined in the Workspace the user is signed in with.

Default Level Use – This check reports any elements on the Default level. (See Suggested Workflow below for more details concerning this check.)

Cell Level(s) – All cells in the Basemapping dgnlib were designed to have the same name as the level they appear on. This checks all cells to make sure their name reflects the level they are on.

Orphan Cell(s) – Reports any and all Orphan Cells (unnamed cells) that are in the file. (See Suggested Workflow below for more details concerning this check.)

Check Pipe Level/Text – All pipe text placed using our automated programs should be on the particular pipe text level and contain the correct text value such as 24” CMP. This checks those levels to ensure their proper level and text values.

Check Text Size/Font – There is a csv file containing all the proper sizes and fonts for those text element placed by our automated programs. This checks the existing text against that csv file for the levels in the Basemapping dgnlib currently, but we can add other dgnlibs levels in as well.

Check NonCell Level – This checks to make sure that the only Element Type of cell are on any level ending with “CELL”.

To the left side again there are two check boxes named Hilite Element and Center Element. These are used to move to each individual element in the list and to hilite it for easier identification.

The Element Information is the large area located across the bottom of the dialog and is used to show the level name, cell name, font and symbology of the selected element.

Updated April 2017

A new revision of NC_Check included the function to create a report file. The button for this report is below the Scan button. This report file is basic in nature, just a listing of the exceptions found for each category along with a Remarks file to allow the user to explain the exceptions.

Basic Suggested Workflow for NC_Check

When you first open the NC_Check, first off deselect all of the Checks located in the middle right section of the dialog. One at a time, select one of the Checks and press the Scan button to the left.

It is important to note that there may be a very large amount of elements reported under several of these checks, which is due to the fact that it reports every element separately. So if you are in a file that has all of the Edge of Pavement on the wrong level, you will see every section listed. It is best to move from element to element by first selecting the Center Element and the Hilite Element check boxes, and then move along from the first element in the list window to each one shown to see what is incorrect. If this is check you are running on a file of your creation, you can stop and fix large scale issue, utilizing normal Microstation functions to fix them globally and then re-scan the file. The large area at the bottom of the dialog can help show what characteristics or symbology is incorrect.

The Check named Default Level Use was put here because during the migration to named levels it was agreed that there should be a level in the dgnlib file for every possible element. The Default level would then be used to detect errors, primarily since copying or merging a file into the active dgn would push elements to the Default level if the proper level was not available through the configuration settings. These elements need to be corrected and if done completely the Default level should be empty.

Cell Levels checks the name of the cell against the name of the level it is on to make sure that they match, minus the word “CELL” at the end of the level name. This does have one exception and that is the Cogo Point Cell at the vertices of all the Exist Structures Minor Pipes RCP, CMP CMAP, HDPE features. This is due to the fact there was so many pipe levels with individual sizes for each, that a single Cogo cell with just the material type was used for all sizes. This will affect only those files that survey import methods were used and the cogo points do not follow dgn into the Final Survey drawing. This exception was written into the program so as to not be listed as an error.

Orphan cells were first put into NC_Check for programming reasons, because it was assumed that we did not use them in our normal workflows. However any cell, such as a tree, that gets clipped during the sheet creation now becomes an Orphan cell. These are not errors, per se, but when checking cells programmatically they do need to be accounted for.

Check Pipe Level/Text makes sure that all Structure Minor Pipe text is on the correct level for the size of the pipe. Too many drawings are coming in with text copied from one pipe to another... without making sure the pipe size/level/text is correct. This check also makes sure the correct size is depicted in the text string itself.

Check Text Size/Font utilizes an external CSV file which lists the font and size for text used in the Basemapping dgnlib.

Check NonCell Level ensures that only cell Element Types are on levels ending in "CELL". In other words, there are no lines, curves, etc on CELL levels.

Lastly, would be to generate a Report file. NC_Check was not designed as a program that can be run, followed by corrections, until there are no exceptions in the file. Some of the categories can be corrected until there are no exceptions, but some may not. Features such as Orphan Cells exist in multiple files due to the clipping functions in Microstation, but do make the file incorrect. NC_Check is written to show all the exceptions, but prints a simple Report in order to itemize those exceptions with possible explanations.

Programmatic Exclusions

It became necessary to exclude some systematic features in order to make NC_Check more useful. Those are as follows:

The Cogo point cells shown on the end points of the pipes/culverts are excluded from the Cell Level(s) category. All the pipes have their sizes written into their level name, but the Cogo points do not. There is only separate pipe material types for the Cogo points and would show multiple exceptions in this category in NC_Check and was thus written out of the program.

All text placed by Geopak do not allow the assignment of any Linestyle to them...let alone the Bylevel attribute. Thus every occurrence of Geopak placed text, such as Cogo elevation, would be shown as an exception in NC_Check. The program was written to ignore their linestyle.

The inclusion of a <Enter> into a pipe size label created utilizing AutoText, in order to make the label into a shorter multiple lines label, caused the Check Pipe Text/Level category to have numerous exceptions. The program was written now to ignore the <Enters> prior to running the check.