

Automatic Tag Reports

by Mark Stefanchuk, Cadgurus.com

Speed, its all about speed. “The faster our computers compute the more time we will have for ourselves.” Someone must have said that, Intel’s Andy Grove perhaps?

I’m not sure it’s worked out that way, but we are still demanding faster and faster computers so we can draw, model, and render faster. And don’t forget about bandwidth and the speed of data transfer.

You and I, we don’t really have any control over either of these situations, processing and bandwidth, well, I mean other than buying the latest available tech. But here is something I can control, the amount of time my computer processes during the day and it doesn’t have to stop when I go home.

If I can get my computer to run a MicroStation process for me, automatically, and do it when I’m not in the room, then the computer is really saving me time. There are several things you can automate in MicroStation like standards audits, file repairs, and plotting.

To me automating design related tasks is more exciting because changes in design workflow impact many people.

Ok, then let’s look at a cadgurus.com macro that extracts tag data to a comma-separated file, and then look at ways to automatically update the output.

tips in here

MicroStation Tags

Most elements in MicroStation are smart including information about position, size, and color. Tags in MicroStation are tools that allow us to attach extra attribute data to elements. Using tags, I can add design data such as manufacturer, model number, or maintenance data like work order, and service date.

Tags are a quick and easy way to link design decisions to MicroStation elements. And, since they remain with the design file element detachment from design data is not a problem as it might be with a linked spreadsheet or database.

Tags however, have some issues. In MicroStation 95 tag definitions could not be exported or imported so transporting tags defined in one file to another wasn’t possible. If you wanted to use tags, the definitions had to be made in the seed file and propagated throughout a project by using the seed file to create new files.

Since MicroStation SE import and export features for tags have been included along with some reporting features. Also realize that MicroStation’s tag-editing tools are simple, and hey, don’t expect design data validation either. Tags are a simply a quick easy way to attach more information to an element.

How do tags work?

Tags are easy to use and set up. First the tag definitions need to be created for the extra attributes you want to track. Here is an example.

- Open the tag definition by selecting *Element>Tags>Define*. This action will display the Tag Sets dialog box.

Tags are the attributes (model, manufacture) and sets are the categories (pumps, valves, compressors) of attributes that will be attached to an element. Start the definition by clicking the Add button under Sets. Notice that the Tag button is grayed until at least one Set has been defined.

- Clicking Add displays a new dialog called Tag Set Name. Type in a name and click **OK** to add the new Set.

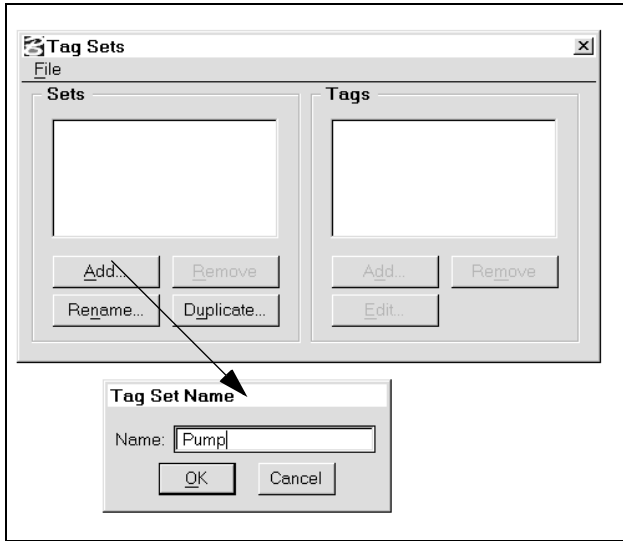


Figure 1 Tag Set Name Dialog

- Next use the Add button under Tags. A new dialog, Define Tags, is displayed. The first tag will be an equipment ID number. Fill in the data as shown in Figure 2 and click **OK**.

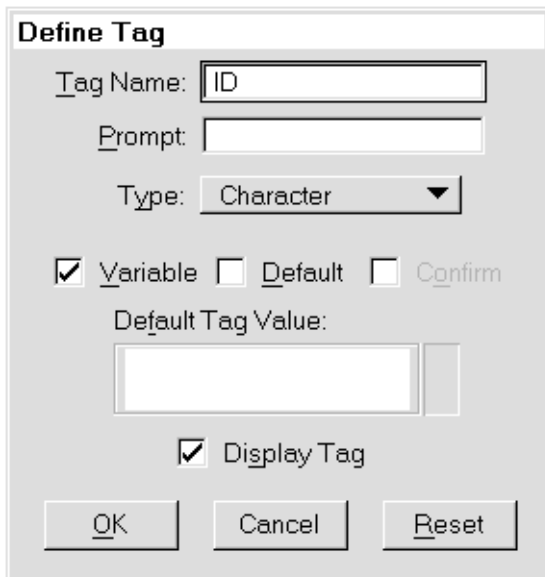


Figure 2 Define Tag Dialog

Cancel will dismiss the dialog without adding the Tag and Reset clears each entry or sets options back to their default values.

Next, for demonstration purposes I have added a tag for service date and work order number. All of the tags were defined as character values so that they could contain both letters and numbers. Of course, the pump tags can be extended to include any number of attributes that you need to track on your pumps.



Figure 3 Pump Tags Defined

To add the tag to the pump graphics, select the Attach Tag command and click on the graphics you want to apply the tag to. Figure 4 shows the Tag Command Toolbox with Attach Tags Command Selected.

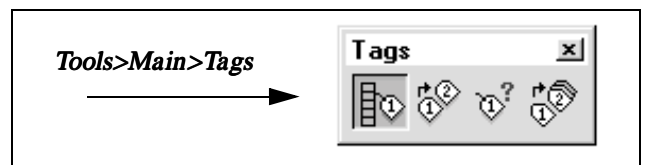


Figure 4 Tag Command Toolbox

The Attach Tags dialog will display. Turn off the display toggle for the service date and the work order tag. Fill in the data for each tag. See Figure 5 for an example.

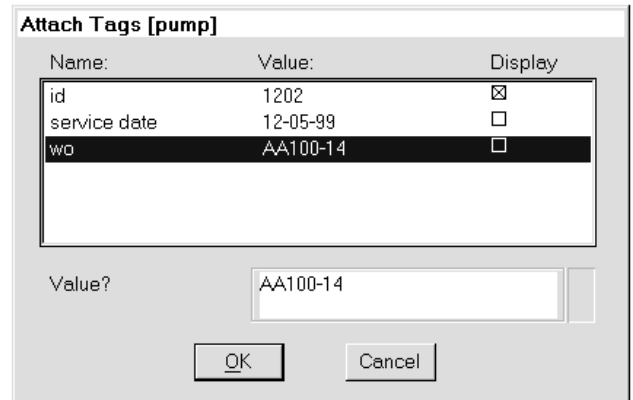


Figure 5 Attach Tags Dialog

- Click **Ok**. At this stage you will be prompted to place the pump ID tag. When done, a drawing may look similar to Figure 6.

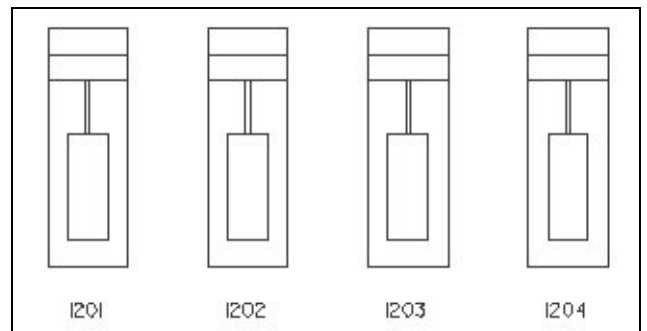


Figure 6 Plan View with Pump Tags

All pumps can be assigned a pump tag with an ID using the MicroStation Tag commands.

Extracting the Data

MicroStation provides some tools for reporting tags but the report format includes "" marks around each character attribute. Further, the delivered report tools do not lend themselves to automation.

As an alternative cadgurus.com has a MicroStation macro called `tagoutvb.bas`. The macro creates a comma-separated file listing each set attachment and the value assigned to each tag in the set. A report based on our pump example is shown in Figure 7.

```
pump,1201,12-05-99,AA-1000-00
pump,1202,12-05-99,
pump,1203,12-05-99,
pump,1204,12-05-99,
pump,1205,12-05-99,AA-1000-98
pump,1206,12-05-99,
pump,1207,12-05-99,
```

Figure 7 Sample Output from Tagoutvb.bas

Creating the Report

MicroStation's BASIC macros can automate tasks by running them on an as-needed basis. Click the command and let the application run. The command doesn't have to be started from MicroStation. A visual basic script can start MicroStation, execute MicroStation commands, and close the application when finished. Here's an example.

If you are comfortable programming MicroStation BASIC then visual basic script will be a natural extension. The following script starts MicroStation and opens files located in `prjdir`.

```
prjdir = "C:\tmp\"
Set msApp = CreateObject("MicroStation.Application")

Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.getFolder(prjdir)
Set fc = f.files

For Each fl In fc
  If InStr(1, fl.Name, ".dgn", vbTextCompare) > 0 Then
    msApp.mbeSendCommand ("rd=" + prjdir + fl.Name)
```

```
end if
next
msapp.quit
```

Don't worry about the details of this script. There are only two things that are really important. First, the second line defines the object `msapp`. This is just a reference to MicroStation. This same line also starts MicroStation. Secondly, the script reads the files in a directory, `prjdir`, and opens the files that have a `.dgn` extension.

The function `msapp.mbeSendCommand` is the same as `mbeSendCommand` found in MicroStation BASIC. In fact, it is MicroStation Basic code. It includes `msapp` so that Windows knows which application the function comes from and which application is needed to execute the command.

`mbeSendCommand` is used instead of `mbeSendKeyin` because MicroStation/J does not recognize `mbeSendKeyin` when the macro is executed using visual basic scripts. In this case the MicroStation executable, `ustation.exe` will launch but the files will not open. The same problem occurs with visual basic executables. Using `mbeSendCommand` takes care of this problem.

So now all we need to do is add the code from `tagoutvb.bas` to the script. Not so fast. Tags are extracted as an array. Unfortunately not all MBE types support arrays when called using visual basic script. Since `tagoutvb.bas` requires a tag array adding the code to a script won't work. To get around this obstacle the script will launch the `tagoutvb.bas` using `mbeSendCommand`.

The following code (on page 31) includes the line `msapp.mbeSendCommand ("macro tagoutvb" + outFile)` plus some additional code to create report headers for each design file read.

A word of caution

Many virus detection software programs make the assumption that visual basic scripts include a computer virus. If this is the case in your office then use Visual Basic 6 or similar to compile the script. The same code, inserted into a header can be compiled to a Visual Basic executable and will not be considered a virus.

The script, the compiled Visual Basic `tagout.exe`, and the `tagoutvb.bas` source are available on cadgurus.com.



```

prjdir = "C:\tmp\"
outfile = prjdir + "tout1.txt"

Set msApp = CreateObject("MicroStation.Application")
Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.getFolder(prjdir)
Set fc = f.files

'--- CREATE AN OUTPUT FILE SO THAT WE KNOW IT IS EMPTY
Set tfo = CreateObject("Scripting.FileSystemObject")
Set tf = tfo.CreateTextFile(outfile, True)
tf.WriteLine(" ")
tf.WriteLine(" CURRENT MAINTENANCE REPORT ")
tf.WriteLine(" ")
tf.close

For Each f1 In fc
  If InStr(1, f1.Name, ".dgn", vbTextCompare) > 0 Then
    msApp.mbeSendCommand ("rd=" + prjdir + f1.Name)

'--- SHOW THE NAME OF THE DESIGN FILE
Set tfo = CreateObject("Scripting.FileSystemObject")
Set tf = tfo.OpenTextFile(outfile, 8)
tf.WriteLine(" ")
tf.WriteLine("DESIGN FILE: " + prjdir + f1.Name)
tf.WriteLine(" ")
tf.close

'--- RUN THE MACRO THAT READS TAGS
msApp.mbeSendCommand ("macro tagoutvb " + outfile)
end if
next
msApp.quit

```

Scheduling a Task

The automation tools require one more step to become completely independent. Using Microsoft's Schedule Task tools the visual basic script can be started at any time and at any frequency. Setting up a scheduled task for 2am every morning simply requires following the steps presented in a Microsoft wizard. In Windows 2000 "Scheduled Tasks" is available in the control panel. Try it!

You can take the example further using the resulting csv file in a spreadsheet or web page. Implementing error checking might lead us into discussions regarding autonomous repair versus splash screen notification. But these topics are beyond our current scope, so perhaps in a future article?

But hey with tags and design data you have lots to work with so don't stop with the simple example used here. Automation can be applied to any repetitive task.

Whether it's counting lampposts, or changing the color of a fan as the maintenance dates pass, put your computer to work while you're off doing something more fun.

About The Author

Mark Stefanchuk is a partner with Ramsey Systems, Inc., the developers of cadgurus.com. Mark can be contacted by email on mark@cadgurus.com.

Please email Mark with any feedback or suggestions for future articles.

CAD