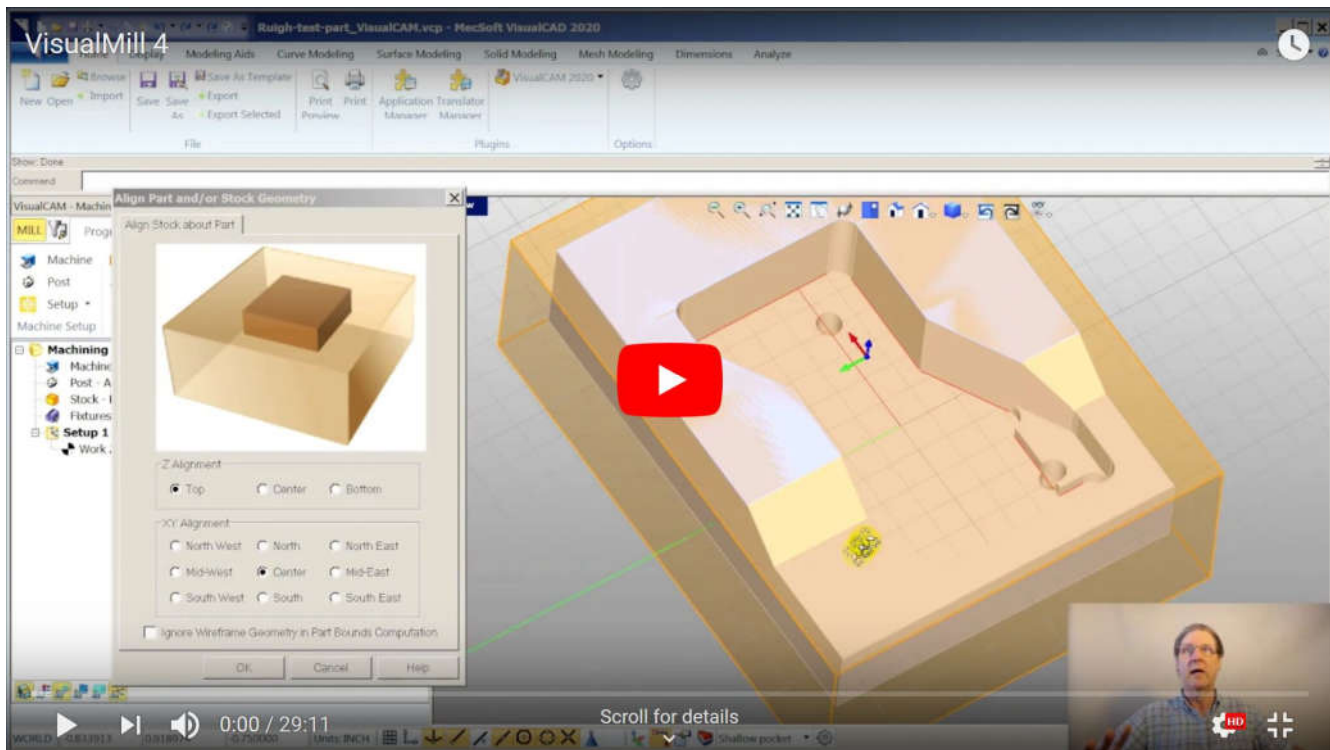




Rako Studios » Media » Suffering-with-software » VisualMill fourth impression (youtu.be/_I_MAph8jrw)

VisualMill fourth impression (youtu.be/_I_MAph8jrw)

VisualMill is a CAM software that runs standalone and inside SolidWorks. I try the stand-alone version in this episode.



SolidWorks part file here.
VisualMill part file here.
SAT, STEP, and .x_t files.

After learning the program in the SolidWorks internal version, I tried the external version. While I like the general user interface and methods of VisualMill, this stand-alone version is not stable, like the internal version. It changes dialog box data I type in, and crashes periodically. I assume this is because the installation failed since I already have a newer version of a Visual C++ than the one it wanted to install.

This version cannot open Solidworks files directly, so it does not save inside those files, it creates a .vcp file. This can make for configuration management problems as you try to figure out what .vcp file comes from what version of your .SLDPRT file.

In addition to stability and multiple file problems, this package was not able to create a proper arc when trying to "convert entities" like you do in SolidWorks. The arc seemed mirrored and even then, not the right shape. This is a deal-killer for me. All the drawing tools seem a bit primitive.

In this episode, rather than do the operations in order, I jumped ahead to where I had to make a sketch so the toolpath would be correct doing the shallow pocket at the bottom. Making that sketch took 20 seconds in SolidWorks, here it was more like 10 minutes.

Part of the low productivity was the fact that I have never used the drawing stuff before. Still, I used AutoCAD for many years, and still use TurboCAD, and between those and SolidWorks, doing a simple sketch should not be this hard.

While the configuration management issues would be a hassle, the BobCAD salesman told me that some folks like a stand-alone package since you don't have to run it on the SolidWorks computer, you can have it on a cheaper or less powerful, or more convenient to the milling machine computer.

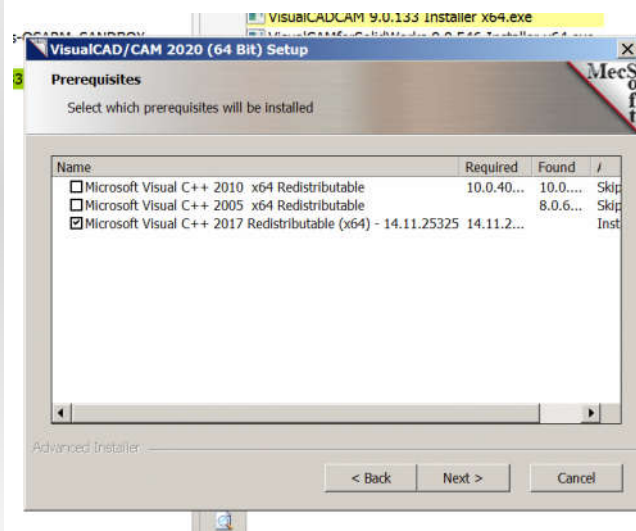
I suppose another solution would be a stand-alone program that would open and save to SolidWorks files, but I am not sure any CAM (computer-aided manufacturing) program that can do that.

I have started to look at BobCAD CAM, and had some trouble getting used to its wizard style interface, and the fact that the simulations seem to run in a patched-in window instead of into the SolidWorks model, the way SW CAM by CAMWorks does. It took me a half-hour to figure out that what BobCAD calls "Pattern" really means "What side of the geometry to cut." I was astonished that it chose to hog through the part model, rather than to the perimeter, I have to be leery of a program that lacks so much "model awareness".

CAMWorks, VisualMill, and BobCAD all have pluses and minuses, but I will reserve judgement until I get more time under my belt.

In one of the practice runs for this video, I did try the "Automatic Feature Recognition," and unlike in the SolidWorks version of VisualMill, this version produced results. I did not let the automatic machining happen, but that might be a fun video, also to see why the SolidWorks version did not find anything, perhaps because I already had toolpaths defined by hand.

I am not a fan of software developed with Microsoft tools, like Visual C++, or Visual .NET, or Visual Basic, for that matter. Microsoft promises how easy it will be while hiding all the horrible bugs that you can't figure out. Just the fact that VisualMill needs three separate Visual C++ installations bothers me:



They call out a 2005, a 2010, and a 2017 version. I assume that is because they never re-wrote the code when Microsoft changed the tool. Microsoft is famous for this, where an update will break something that used to work. I assume that is why VisualMill is unstable on my machine, since it has a newer C++ than the one VisualMill expects.

I worry about software based on crappy Microsoft design practices, but it seems like many CAM packages use this C++ package.