



## Version Control of G-Code

Version control of G-code has become a critical issue in many industries, including medical, defense and aerospace. OEMs have been known to shut down the operations of suppliers who can not adequately enforce a regimen of version control that extends all the way from design to G-code. Manufacturers have found that maintaining version controls on G-code is not easy. Failure to maintain control can have disastrous consequences.

In most manufacturing companies version control of documents, drawings, models and simulations related to a part or product is handled by a PDM application. PDM applications place related information into a vault and control access to the information by granting or denying permissions to read or check out data for maintenance. G-code is usually not included in this scheme because access is needed on the shop floor and most shop floor personnel are not users able to check data out of the vault. In today's modern shops, most G-code programs are requested by a machine operator through a control and sent to the control, often through a serial port. For a host of good reasons, most machine tool controls are not part of company Ethernet networks. This creates issues for PDM applications. Large G-code programs are often drip fed to controls, another issue for PDM applications.

Version control of G-code is impossible if the diskette drives on machine tool controls remain working. Walking around a shop and seeing a stack of diskettes on each machine operator's toolbox is a sure sign that there is no version control of G-code in the shop. Companies who control G-code tend to disable diskette drives, USB interfaces and other external media interfaces on machine tool controls. These companies also tend to keep just one program in memory of the machine tool. With Fanuc and other controls, parameter settings allow users to delete all programs every time a new program is loaded.

To insure that only the most current version of the part is being machined, some companies have adopted a policy of not keeping G-code files. They have purchased additional seats of CAM software and post programs each time one is needed. The PDM application can control the CAM file. This allows manufacturer to insure that only the most current version of the part is being produced. While CAM vendors rejoice when companies take this track, it is an expensive path to choose.

Modern DNC applications allow users to maintain version control using a file folder system. Typical DNC file folders have a folder for each machine. Inside that folder are at least three (3) other folders; Approved for Download; Under Development; and Upload. The Approved for Download folder contains G-code programs which have been approved and are available for use. These can be sent to the machine requesting the file.



# reThink Engineering, Inc.

---

5554 Eureka Drive \* Suite I \* Hamilton, OH 45011 \* (513) 737-6408 Phone \* (513) 737-6412 Fax \*  
support@rethinkeng.com

Programs in the Under Development folder are files for new versions which have not been approved for production yet. These can not be sent to machine controls without a supervisor's approval. The programs in the Uploads folder have been modified on the floor and sent back to the system for review. These can not be sent to the machine control without a supervisor's approval. A DNC application can typically be set up to look for the program being called in the Uploads and Under Development folders before looking in the Approved for Download folder. If a program is found in either the Uploads or the Under Development folders, the system can send a message to the machine operator indicating the finding and suggesting the operator see a supervisor. The supervisor can then make a decision to either approve the program or to send an older version to the machine. Under this scenario, the approval of a program adds either a comment to the G-code itself or adds a line to the history file for the program indicating who approved it and when. Ultimately, this scenario takes the operator out of the decision-making process and flags potential version control issues.

Strengthening the folder scenario in many companies is a routine in the company CAM system that automatically posts all CAM files that are saved and places the resulting G-code into the Under Development folder for the machine tool. As engineering changes occur, this automated posting routine insures that new revisions of each part are propagated throughout the system, all the way to the G-code. Many CAM applications have this as an option and others have tools that allow users to automate the process.

Over time, PDM applications will figure out how to control G-code. These systems are already being tightly integrated with CAD/CAM applications. The integrated suites like CATIA and Unigraphics have largely ignored G-code. There is hope that users will force these companies to do a better job. TopSolid is launching a newly rewritten suite of CAD/CAM applications that are built on a PDM base. They have released the CAD portion, but are still a few months from the CAM release and haven't given any indications about G-code support within PDM. Until users from the industries where version control of G-code make their needs known, vendors of CAM, DNC and PDM will ignore the issue. If version control of G-code is important, make your software vendors aware of the issue.

**About the Author:** Kurt Freimuth is the President of reThink Engineering, Inc. a value-added reseller of GibbsCAM and TopSolid software. reThink Engineering is located in Hamilton, OH, just north of Cincinnati.