

Pro/ENGINEER Launch Scripts

Table of Contents

Pro/ENGINEER Launch Scripts.....	1
Table of Contents.....	1
Why a Pro/ENGINEER launch script?	2
Scripting Language.....	2
Accessing a Command Prompt:.....	2
Getting Help on Command Line Scripting	2
Comment your scripts.....	2
Windows Resource Kit	3
Additional commands.....	3
Launch Script Snippets	3
Launch Script Framework.....	3
Multiple Pro/ENGINEER Releases	4
Single Script Framework.....	4
Multiple Script Framework.....	5
Pro/ENGINEER Load Point Variance.....	6
Startup Working Directory	6
Local Cache Configuration	6
ProSystem Network Drive	7
Startup Working Directory Cleanup.....	8
Trail File Management	9
Company Standards.....	9
Config.pro Mappings.....	10
License Selection.....	11
Command Line Option	13
Windows User Name.....	13
Limit the PSF files	14
Bringing it all together	15
proestart.bat	15
proestart3.bat.....	16
proestart5.bat.....	17
mapdrive.bat	18
clean.bat.....	19
cmdlineargs.bat	19
setlicense.bat.....	20
Summary	21

Why a Pro/ENGINEER launch script?

PTC products are capable of running when they are installed out of the box so why create a batch file to start Pro/ENGINEER? PTC's installation approach to Pro/ENGINEER is a single-user desktop application. When the installer finishes a user local to that machine can run Pro/ENGINEER. The out of the box install does not take into consideration your company's standards or the system administration overhead for managing multiple Pro/ENGINEER users built into the installer.

All Pro/ENGINEER users should adhere to the company standards. These standards include anything and everything that assists in configuring the Pro/ENGINEER environment, such as templates and libraries, configuration files, even the Pro/ENGINEER launch script.

Uniformity simplifies end-user support. Knowing the common environment and expected behavior makes it easier to identify issues. Finding the source of an issue is easier when all Pro/ENGINEER users have the same configuration.

The administrator must be able to enforce company standards. Standards located in a single, common location on the network or in the data management system are much easier to administrate than standards located on each user's computer.

Scripting Language

Practically any scripting language (VBS, VisualBasic, Java, ANT, etc) can be used providing it can copy/delete files and launch programs. What is needed is a scripting language that is easy to use and understand and supported across all Windows platforms without any special installation procedure or editor. The winner is Windows command line options (a.k.a DOS).

Batch files (*.bat) or command files (*.cmd) are both ascii text files that use the same DOS commands. It doesn't matter which file extension you use.

Accessing a Command Prompt:

The command prompt can be accessed by selecting Start > Run and typing in "cmd.exe" > Enter. Or you can go to Start > Programs > Accessories > Command Prompt (Windows 2000). Either method starts a Command Prompt at the root directory ("C:\").

A windows utility named "Command Prompt Here" can also be used to start a Command Window in any specified directory. It works through all Windows versions. The installation program is called "doshere.inf". It can be found in the zip file along with this document. Simply right click on this file and select install. After it installs, right click on any folder in Windows Explorer and choose "Command Prompt Here" to use it.

Getting Help on Command Line Scripting

Help for command line scripting is everywhere. From a cmd shell, type "help". Search windows help and support system for "command line". When looking for techniques or more than just a command, use Google.

Comment your scripts

ALWAYS COMMENT YOUR SCRIPT FILES!!! Memories fail and you may not be the next person to review and edit a script. Comments allow you and others to quickly determine why the script is written in a particular manner.

There are two types of comments as shown below. The double colons (“::”) are typically used to designate comments. The asterisks are just used as a method to denote a comment for the user. They are not necessary. In early Windows XP releases the double colons didn’t always prevent the line from executing. So be careful using “::” to disable a command.

```
:: Map network drive  
rem net use P: \\ptc_server\c$\ptc\ProSystem
```

The “REM” statement is the original DOS command used to comment out a line. It can be in upper or lower case and it can be used to temporarily disable a command as shown below.

Windows Resource Kit

Microsoft produces Windows Resource Kit Tools for each Windows release. It includes additional commands that can be run from the command prompt. While none of these are used in this document, it is worth noting their existence for other scripting applications.

Additional commands

Many programs provide “command line” interfaces. Any of these commands that can be executed from a command prompt can be included in a batch file. Applications such as WinZip provide downloadable interfaces for command line execution.

Launch Script Snippets

Launch scripts are composed of generic components but are always environment specific. A practical approach to understanding a launch script is to discuss the building blocks and some common use cases.

The use cases will use a fictitious company called ABC Corporation.

Launch Script Framework

The simplest launch script is a batch file that calls a Pro/ENGINEER launch command. This example is the framework for adding launch script snippets.

```
@echo off  
:: proestart.bat  
:: Pro/ENGINEER Startup Script  
:: Created 06/01/2010  
:: By Matt Meadows  
:: Solutions Architect  
:: VIRSO Inc  
  
:: Set Environment Variables  
set proe_load_point=D:\ptc\proe\proeWildfire3.0  
  
:: Launch Pro/ENGINEER  
call %proe_load_point%\bin\proe.exe  
  
@echo on  
exit
```

Note how the Pro/ENGINEER load point was turned into a variable in the script. This technique allows common script modifications to be centralized in a small section of the script (i.e. the environment variables).

When executed, proe.exe will identify all *.psf files in the Pro/ENGINEER load point and will prompt the user to choose a license configuration if more than one PSF file exists.

Multiple Pro/ENGINEER Releases

ABC Corporation is ready to beta test Pro/ENGINEER Wildfire 5.0. A small group of beta testers have been identified and Pro/ENGINEER Wildfire 5.0 has been installed on their workstations along with Pro/ENGINEER Wildfire 3.0.

There are a couple of methods for addressing this use case. All possible solutions can be maintained in a single launch script or the launch script can be split into separate 3.0 and 5.0 scripts called from a common framework. Both techniques are shown.

Single Script Framework

A single script file is easy to track. All changes occur to a single file and there are no file interdependencies to worry about.

```
@echo off
:: proestart.bat
:: ...

:: Set environment variables
Set proe_load_point=D:\ptc\proeWildfire

:: Set Pro/ENGINEER release to run if specified
set release=%1
:: Ask for Pro/ENGINEER command to use if not specified.
if not defined release (
  echo 3.0 = Wildfire 3.0
  echo 5.0 = Wildfire 5.0
  echo ...
  set /P release="What release would you like to run (3.0 or 5.0)?"
)

:: Start Pro/ENGINEER
if %release%==3.0 (
  :: Configure environment for Pro/ENGINEER Wildfire 3.0
  :: ...
)
if %release%==5.0 (
  :: Configure environment for Pro/ENGINEER Wildfire 5.0
  :: ...
)
call %proe_load_point%%release%\bin\proe.exe

@echo on
exit
```

Note the proe_load_point and the release number were split into separate variables. This technique only works if the installation folders are in parallel installation directories: D:\ptc\proeWildfire3.0 and D:\ptc\proeWildfire5.0.

The Pro/ENGINEER launch script can take a command line option that calls out releases to use. For non-beta testers, ensure their desktop shortcuts call out the version 3.0 (e.g. start in: P:\bin\prestart.bat 3.0). These users won't notice any difference in the way Pro/ENGINEER launches.

For the beta testers, remove the version from the desktop shortcut (e.g. start in: P:\bin\prestart.bat). When the release isn't called out, the script will prompt the user to choose a release.

Multiple Script Framework

The separate script files example is a preferred approach because it isolates changes to the 5.0 script from the 3.0 script. There are three batch files involved in this example: proestart.bat, proestart3.bat, and proestart5.bat. The separate scripts make it easier to remove the obsolete proestart3.bat script after the move to Pro/ENGINEER Wildfire 5.0 is complete.

proestart.bat

```
@echo off
:: proestart.bat
:: ...

:: Set environment variables common to all releases
Set proe_load_point=D:\ptc\proeWildfire

:: Set Pro/ENGINEER release to run if specified
set release=%1
:: Ask for Pro/ENGINEER command to use if not specified.
if not defined release (
    echo 3.0 = Wildfire 3.0
    echo 5.0 = Wildfire 5.0
    echo ...
    set /P release="What release would you like to run (3.0 or 5.0)?"
)

:: Execute common tasks here
:: ...

:: Start Pro/ENGINEER
if %release%==3.0 (
    call proestart3.bat
)
if %release%==5.0 (
    call proestart5.bat
)

@echo on
exit
```

Any tasks that are common across all releases should be executed in this script prior to calling the proestart3.bat or proestart5.bat scripts.

```
@echo off
:: proestart3.bat
:: Configure the Pro/ENGINEER Wildfire 3.0 environment
:: ...

:: Launch Pro/ENGINEER Wildfire 3.0
call %proe_load_point%3.0\bin\proe.exe

@echo on
exit
```

Note the environment variable proe_load_point is automatically passed from one batch script to the next. This is standard functionality of the "call" statement. Any tasks explicit to 3.0 should be included in this script.

```

@echo off
:: proestart5.bat
:: Configure the Pro/ENGINEER Wildfire 3.0 environment
:: ...

:: Launch Pro/ENGINEER Wildfire 3.0
call %proe_load_point%3.0\bin\proe.exe

@echo on
exit

```

Any tasks explicit to 5.0 should be included in this script. How the scripts are split depends on the requirements.

Pro/ENGINEER Load Point Variance

Sometimes applications are installed on non-default drives due to disk space issues and hardware non-uniformity. For example, ABC Corporation has some workstations with only a C drive and others with C and D drives. The preferred installation location is D drive when available. This script snippet finds the Pro/ENGINEER load point. If installations exist on C and D drives, the script defaults to the D drive.

```

:: Set the text editor command for use in Pro/ENGINEER
if exist C:\ptc\proeWildfire3.0\bin\proe.exe (set proe_drive=C)
if exist D:\ptc\proeWildfire3.0\bin\proe.exe (set proe_drive=D)
set proe_load_point=%proe_drive%\ptc\proeWildfire3.0
:: ...

:: Launch Pro/ENGINEER Wildfire 3.0
call %proe_load_point%\bin\proe.exe

```

Note, environment variables may also be set in the <Pro/ENGINEER load point>\bin*.psf files. However, these files are overwritten when PTCSetup is executed. Thus it is recommended not to modify the *.psf files and to keep the environment variables in the launch script.

Startup Working Directory

ABC Corporation is concerned about users changing the startup working directory for Pro/ENGINEER. They want to override the shortcut start in location and enforce a common startup working directory for all users. If the location does not exist, create it.

```

:: Set Environment Variables
set proe_work_dir=D:\workdir3.0\

:: Set the startup working directory
if not exist %proe_work_dir%\ (mkdir %proe_work_dir%)
cd /d %proe_work_dir%

```

Note the working directory is explicit to Pro/ENGINEER Wildfire 3.0 (e.g. “workdir3.0”). Different Pro/ENGINEER releases installed on the same machine at the same time need separate startup working directories.

Local Cache Configuration

The Pro/ENGINEER users may switch workstations and some of them have multiple logins to Windchill. Each Windchill user login needs a separate local cache structure to avoid local cache corruption.

```

:: In proestart.bat
:: Request Windchill username
set wcusername=%2

```

```

:: Ask for Pro/ENGINEER command to use if not specified.
if not defined wcusername (set /P wcusername="What is your Windchill username?")

:: In prostart3.bat
:: Set Environment Variables
set proe_work_dir=D:\workdir3.0
:: Set the Windchill local cache location per the logged in user
set ptc_wf_root=%proe_work_dir%\%wcusername%\wf

:: Create the local cache folder if it doesn't exist
if not exist %ptc_wf_root% (mkdir %ptc_wf_root%)

:: Set the server registry file
robocopy /mir %pro_system_dir%\configs\config.fld %ptc_wf_root%\Settings\config.fld

```

This script uses the startup working directory as the root folder for the local cache.

A common local cache conflict occurs when two separate Windchill users log in to the same Windchill server using the same Pro/ENGINEER installation. For example, a user logs into Windchill through Pro/ENGINEER using two different accounts: "bob_user" and "bob_admin". As bob_user, he creates a workspace named "test", does some work and closes Pro/ENGINEER. Then he launches Pro/ENGINEER again as bob_admin and accesses the "test" workspace. Generally, this will result in some level of data corruption in the local cache. To avoid corruption, make the local cache location Windchill user specific (not just machine login specific).

Every Pro/ENGINEER user must register to the Windchill server using the same manually entered server name. This information is stored in the <ptc wf root>\Settings\config.fld file in the local cache directory structure. Copying this file between installations will automatically register the server for the end users and avoid any typing mistakes.

ProSystem Network Drive

Recall in the original environment, a shared network location named "ProSystem" must be mapped as a local drive named "P:\". This script must ensure the drive mapping exists and that it is correct. If it is mapped to an inappropriate location, remap it using the following script snippet.

```

:: Map Network Drive
if exist P:\ (
  if not exist P:\admin\prostart.bat (net use P: /delete)
)
if not exist P:\ (net use P: \\ptc_server\ProSystem)

```

Adding a little complexity, the ProSystem shared location may be on the local machine. Note Pro/ENGINEER is not tested or supported on ANY Windows server operating system. This only makes sense if a Pro/ENGINEER workstation hosts the standards or when a user has a local copy of the standards (i.e. off-line users).

```

:: mapdrive.bat
:: Map Network Drive
:: Set Environment Variables
set pro_system_drive=%1
set pro_system_share=%2
set pro_server=%3
set pro_server_path=%4

:: Decision Logic
:: 1) Is this the drive mapping correct?

```

```

:: A) No: Does the drive mapping exist?
:: I) Yes: Is it the server?
:: a) Yes: Remove drive substitution
:: b) No: Remove drive mapping
:: II) No: Map or substitute the drive
:: a) Is it the server?
:: i) Yes: substitute drive
:: ii) No: map drive

if not exist %pro_system_drive%\bin\proestart.bat (
if exist %pro_system_drive% (
if %pro_server%==%computername% (
subst %pro_system_drive% /d
) else (
net use %pro_system_drive% /delete
)
)
if %pro_server%==%computername% (
subst %pro_system_drive% %pro_server_path%
) else (
net use %pro_system_drive% \\%pro_server%\%pro_stds_share% /PERSISTENT:Yes
)
)
)

```

This majority of this script snippet will never change. It is rather long and could be externalized as a separate script called from the main script like this:

```

:: Map Network Drive
call mapdrive.bat P: ProSystem mmeadows01 D:\ptc\ProSystem

```

Startup Working Directory Cleanup

Over time, the Pro/ENGINEER startup working directory can fill up with miscellaneous temporary files. Most of these files are not relevant beyond the current session of Pro/ENGINEER. A useful launch script task is to clean out the temporary file clutter.

```

:: Purge the working directory
call purge.bat

:: Remove logs, and other temporary files
del /Q /F %proe_work_dir%\*.log*
del /Q /F %proe_work_dir%\*.ger*
del /Q /F %proe_work_dir%\*.m_p*
del /Q /F %proe_work_dir%\*.tst*
del /Q /F %proe_work_dir%\*.plt*
del /Q /F %proe_work_dir%\*.err*
del /Q /F %proe_work_dir%\*.out*
del /Q /F %proe_work_dir%\*.ptd*
del /Q /F %proe_work_dir%\*.idx*
del /Q /F %proe_work_dir%\*.lst*
:: Enable deletion of trail files if no one will ever run a trail file.
rem del /Q /F %proe_work_dir%\trail.txt*
:: Note: *.crc files should not be deleted. Circular references should be corrected.
rem del /Q /F %proe_work_dir%\*.crc*

```

This script snippet will remove log and other temporary files every time Pro/ENGINEER is launched. It is easy to understand but can get very long, very quickly. A more programmatic approach that combines startup working directory cleanup and trail file management in a separate script is shown in the “bringing it all together” example at the end of this document.

Trail File Management

Some companies set the trail_dir config.pro option to a specified folder location but the users do not regularly clean out this folder and the trail files build up. Remove trail_dir from the config.pro files and let the launch script clean up the trail files on every launch of Pro/ENGINEER.

Sometimes users want to replay the last trail file to recover partial work. The last trail file needs to be renamed before it can be played back. Use this script snippet to backup the last trail file and prepare it for replay.

```
:: Set Environment Variables
:: Continue From Out Of Sequence
set continuefromoos=1

:: Purge the working directory
call purge.bat

:: Backup the last trail file
copy /y %proe_work_dir%\trail.txt* %ptc_work_dir%\last_trail_backup.txt

:: Remove last trail file
del /Q /F %proe_work_dir%\trail.txt*
```

Trail files cause Pro/ENGINEER to exit when they go out of sequence. The only way to keep Pro/ENGINEER open is to use the continuefromoos environment variable. The trail file may still go out of sequence. But Pro/ENGINEER won't exit. This is a common environment variable for all Pro/ENGINEER releases. It is only useful if a Pro/ENGINEER user needs to replay a trail file.

The deletion of all trail files from the previous script snippet was moved to the end of this script snippet. The purge command will reduce the number of trail files to one, the latest. The xcopy command will rename it so it can be played back.

Note: This snippet may be obsolete with Pro/ENGINEER Wildfire 5.0's session recovery functionality.

Company Standards

ABC Corporation has a choice about where to store most company standards. They can be stored on a network drive (ProSystem) or in PDMLink. A diminishing number of Pro/ENGINEER standards are not supported inside of Windchill. It is a best practice to store as much in Windchill as possible.

A default set of standards is installed with Pro/ENGINEER. It is a good idea to remove any of the default standards that conflict with corporate standards. For example, delete all template models from the <Pro/E load point>\templates folder before copying the corporate standard templates into this folder.

```
:: Remove User's Standards
del /Q /F %proe_load_point%\templates\

:: Add Company Standards
xcopy /q /y %pro_system_dir%\templates\ %proe_load_point%\templates\
xcopy /q /y %pro_system_dir%\configs\config.sup %proe_load_point%\text\config.sup
xcopy /q /y %pro_system_dir%\configs\config.pro %proe_load_point%\text\config.pro
xcopy /q /y %pro_system_dir%\configs\config.win %proe_load_point%\text\config.win
xcopy /q /y %pro_system_dir%\configs\tree.cfg %proe_load_point%\text\tree.cfg
```

The length of this section depends on the scope of the corporate standards: config files, toolkit applications, ModelCHECK integration, etc.

Robocopy comes with the Windows Resource Kit and is an alternative to xcopy. Robocopy executes faster because it analyzes file time stamps and byte counts and doesn't recopy files that have not changed. The mirror option deletes any extra files in the target location and overwrites any files determined to be out of sync (older or newer in the target location).

```
:: Set Environment Variables
Set path= P:\utilities;%PATH%

:: Add Company Standard Template files
robocopy %pro_system_drive%\templates\ %proe_load_point%\templates2\ /mir

:: Add Company Standard Configuration files
robocopy %pro_system_drive%\configs\ %proe_load_point%\text\ /if config.sup,config.win,config.pro,tree.cfg
```

Note: This script snippet requires Robocopy to be installed for each user or made available to all users from a shared network location with execute permissions. Robocopy is often made available to all users as P:\utilities\robocopy.exe. The full path to Robocopy must be specified or the path must be added to the path environment variable.

Be very careful about using the /MIR option as it deletes all files in the target that don't match the source.

If this portion of the launch script becomes too large to manage as a single file Robocopy can use an external job file.

Config.pro Mappings

Pro/ENGINEER can use environment variables to set the values of config.pro and config.sup options. The config.pro options reference environment variables using the convention \$<environment variable>.

In some instances, applications will be installed in different locations from one computer to the next. In these instances, it may be preferable to set system environment variables on the local computer and not in the batch script.

For example, ABC Corporation has three configured instances of the Pro/ENGINEER Interface for UG and the UG installation is not consistent for all users. Set the option's value as a system or user environment variable on the local computer instead of in the Pro/ENGINEER launch script.

If there is only one exception to the standard installation location, it might be best to set the environment variable for that exception in the batch script. For example:

```
:: Set the UG load point
set ug_install_dir=C:\ug
if %computername%=="mmeadows01" (set ug_install_dir=D:\UG)
```

Once the environment variable is set it can be called out in the config.pro file using the \$<environment variable> syntax as shown.

! config.pro

! In config.pro file, set the UG installation directory
intf3d_ug_install_dir \$ug_install_dir

Existing system environment variables can be used.

:: Set the text editor to use with Pro/ENGINEER
set text_editor="%ProgramFiles%\Windows NT\Accessories\wordpad.exe"

In the config.pro file, add the following:

! Set the text editor to use with Pro/ENGINEER
pro_editor_command \$text_editor

License Selection

Pro/ENGINEER *.psf files are created during installation, one for every license configuration. When executed, Pro/ENGINEER's launch command "proe.exe" looks for *.psf files in the <Pro/E load point>\bin directory. If only one is found, that psf file is executed. If more than one is found, proe.exe returns a dialog listing the various psf files to choose from. This behavior can be bypassed by adding the psf file of choice as a command line option. For an example of this syntax, look at <Pro/E load point>\bin\<launch command>.bat where <launch command> is the same name as the psf file (e.g. proe1.bat and proe1.psf).

There are multiple license assignment examples in this document. To avoid confusion, choose one technique the best suits the license/user environment and stick with it.

ABC Corporation has three licensing goals:

- Make it easy on the Pro/ENGINEER users
- Minimize licensing conflicts
- Maximize license utilization

There are multiple approaches to meeting this requirement. The following three methods support ABC's licensing environment.

ABC company has 20 Pro/ENGINEER users: 5 heavy, 13 occasional, and 2 managers. There are only 15 Pro/ENGINEER licenses and the current Pro/ENGINEER license pack consists of:

- 5 Flex3C licenses
- 5 Foundation licenses with AAX
- 5 Foundation licenses

The Flex3C licenses were purchased for the heavy users. Occasional users have a choice of Foundation with AAX or Foundation. The managers are only supposed to use the Foundation licenses. The license configurations are captured in flex3c.psf, found_aax.psf, and foundation.psf files in the <Pro/ENGINEER load point>\bin directory.

It is rare for all licenses to be in use at the same time, but it is common to run out of a particular type of license, especially the Foundation with AAX licenses. The launch script determines licensing as part of the Pro/ENGINEER launch command.

There are two managers that always use the same login on the same computers and only need a foundation license. We can hard code their Pro/ENGINEER launch commands in the script, ensuring they never accidentally take a Foundation with AAX or Flex3C license.

```
:: Force the managers to use the foundation licenses
if %username%==mgr1 (call %proe_load_point%\bin\proe.exe %proe_load_point%\bin\foundation.psf)
if %username%==mgr2 (call %proe_load_point%\bin\proe.exe %proe_load_point%\bin\foundation.psf)
```

There are 5 heavy users, but they don't always use the same computer. The script needs to identify these users based on their Windchill username. The wcusername environment variable was set previously and shown again in this script snippet.

```
:: Request Windchill username
set wcusername=%2
:: Ask for Pro/ENGINEER command to use if not specified.
if not defined wcusername (set /P wcusername="What is your Windchill username?")

:: Identify the heavy users by Windchill login name
Set flex3c_users=heavy1,heavy2,heavy3,heavy4,heavy5

call :parse "%flex3c_users%"
goto :eos

:parse
:: Uses reduction to work through the list of user.
set list=%1
set list=%list:="%
FOR /f "tokens=1* delims=" %%a IN ("%list%") DO (
    if not "%%a" == "" call :sub %%a
    if not "%%b" == "" call :parse "%%b"
)
goto :eos

:sub
:: The meat of the script is here.
set _user=%1
:: Force the heavy users to use the Flex3C licenses
if %wcusername%==%_user% (call %proe_load_point%\bin\proe.exe %proe_load_point%\bin\flex3c.psf)
goto :eos

:eos
endlocal
@echo on
```

This example compares the Windchill username to the list of logged in users. When it finds a match it launches Pro/ENGINEER with the flex3C license option. The script uses reduction to iterate through a list of heavy users of unknown length.

As the number of users grows, it may make sense to externalize the list of users in a list file. The third script example reads from the following list of occasional users to make the comparison.

```
:: occasionalusers.lst
User1
User2
```

User3
User4
User5
User6
User7
User8
User9
User10
User11
User12
User13

Each user in occasionalusers.lst is identified by a carriage return.

```
for /f "eol=: tokens=* delims= " %%i IN (occasionalusers.lst) do (
  :: Is the user an occasional user?
  if %wcusername%==%%i (
    :: Ask the occasional users to choose a license to run
    set license=%3
    :: Ask for a Pro/ENGINEER license to use if not specified.
    if not defined release (
      echo 1 = Foundation
      echo 2 – Foundation with AAX
      echo ...
      set /P release="What license would you like to run (1 or 2)?"
    )

    :: Start Pro/ENGINEER
    if %license%==1 (call %proe_load_point%\bin\proe.exe %proe_load_point%\bin\foundation.psf)
    if %license%==2 (call %proe_load_point%\bin\proe.exe %proe_load_point%\bin\found_aax.psf)
  )
)
```

The launch script checks to see if the license option was passed through the launch command (e.g. "P:\bin\proestart.bat 3.0 User11 2"). If not, it iterates through the list of users looking for a match. When a match is found, the user is prompted to choose a license option.

Command Line Option

As demonstrated, the Pro/ENGINEER launch script can take command line options that call out licenses to use. Modify the shortcut on the desktop for each user and explicitly call out the license to use. This is the same technique as calling out different Pro/ENGINEER releases to run. This technique is discouraged because the user can change or remove the command line option in their desktop shortcut.

Some administrators find this useful because they can include the command-line option in the desktop shortcut and exclude it from the start menu shortcut. This defaults the license but doesn't prevent the user from choosing a different license to run.

Windows User Name

When users log in to the local computer, the %USERNAME% environment variable is set to the user's logon name. This environment variable can be used to choose the license based on the logged in user.

```
:: Force user to use the foundation license based on Windows login name
if %username%==mmeadows (call %proe_load_point%\bin\proe.exe %proe_load_point%\bin\foundation.psf)
```

This solution works well when a user must always use a particular license. For example, contract employees may have purchased Pro/ENGINEER licenses and be required to use their license rather than pulling a corporate license.

Note: In a Windchill environment it is more appropriate to ask the user for their login credentials. This maintains separation of the local cache per the Windchill user and not just the machine user.

Limit the PSF files

Ensure only one psf file is copied into the <Pro/E load point>\bin directory. This guarantees the Pro/ENGINEER users don't pull license configurations they are not supposed to use.

```
:: Make available Foundation and Foundation with AAX licenses based on Windows login name
if %username%==mmeadows (
  del /q /f %proe_load_point%\bin\*.psf
  xcopy /q /y %pro_system_dir%\configs\foundation.psf %proe_load_point%\bin\foundation.psf
  xcopy /q /y %pro_system_dir%\configs\foundation.psf %proe_load_point%\bin\found_aax.psf
)
call %proe_load_point%\bin\proe.exe
```

This practice is great if the user should never be allowed to choose a different license option or if only a sub-set of license options should be available to the user (e.g. foundation and foundation + AAX, but not flex3C).

Bringing it all together

To review, ABC Corporation has a single site using Pro/ENGINEER Wildfire 3.0 with Windchill PDMLink in production. They are beta testing Pro/ENGINEER Wildfire 5.0 on a few workstations. Below are the embedded and documented batch files.

The decision was made to split the launch script to isolate changes between Pro/ENGINEER Wildfire 5.0 and Pro/ENGINEER Wildfire 3.0. This script places only common snippets that are most likely not to change between Pro/ENGINEER releases in proestart.bat. This is to prevent any unforeseen changes from affecting the production environment.

proestart.bat



Proestart.bat

```
@echo off
:: proestart.bat
:: Pro/ENGINEER Startup Script
:: Created 06/01/2010
:: By Matt Meadows
:: Solutions Architect
:: VIRSO Inc

:: Syntax: proestart.bat %release% %wcusername% %license%
:: Where: all command line arguments are optional
:: %release% is one of the available releases (e.g. 3.0 or 5.0)
:: %wcusername% is the Windchill username for this session
:: %license% is the license configuration to use (e.g. foundation)

:: Set environment variables common to all releases
:: Capture the batch files location
set currentpath=%~dp0
:: Load command line arguments
call cmdlineargs.bat %1 %2 %3
:: Continue From Out Of Sequence
set continuefromoos=1
:: Pro/E load point (less the version number)
set proe_load_point=D:\ptc\proeWildfire

:: Request the Pro/ENGINEER command to use if not specified as a command line option.
if not defined release (
  echo 3.0 = Wildfire 3.0
  echo 5.0 = Wildfire 5.0
  echo ...
  set /P release="What release would you like to run (3.0 or 5.0)?"
)

:: Request Windchill username if not specified as a command line option.
if not defined wcusername (set /P wcusername="What is your Windchill username?")

:: Configure for Pro/ENGINEER
if %release%==3.0 (call proestart3.bat)
if %release%==5.0 (call proestart5.bat)

@echo on
exit
```

proestart3.bat



Proestart3.bat

```
:: proestart3.bat
:: proestart3.bat
:: Configure the Pro/ENGINEER Wildfire 3.0 environment

:: Set the text editor command for use in Pro/ENGINEER
if exist C:\ptc\proeWildfire3.0\bin\proe.exe (set proe_drive=C)
if exist D:\ptc\proeWildfire3.0\bin\proe.exe (set proe_drive=D)
set proe_load_point=%proe_drive%\ptc\proeWildfire3.0

:: Set Environment Variables
set proe_work_dir=D:\workdir3.0\
:: Set the Windchill local cache location per the logged in user
set ptc_wf_root=%proe_work_dir%\%wcusername%\l.wf
:: Mapped Drive variables
set pro_system_drive=P:
set pro_system_share=ProSystem
set pro_server=mmeadows01
set pro_server_path=D:\ptc\ProSystem
:: Add Robocopy to the path for the local user.
set path= %pro_system_drive%\utilities;%PATH%
:: Set the UG load point
set ug_install_dir=C:\ug
if %computername%=="mmeadows01" (set ug_install_dir=D:\UG)
:: Set the text editor to use with Pro/ENGINEER
set text_editor=%ProgramFiles%\Windows NT\Accessories\wordpad.exe

:: Group users and licenses
:: Heavy users
set group1_users=heavy1,heavy2,heavy3,heavy4,heavy5
set group1_licenses=flex3c
:: Manager users
set group2_users=mgr1,mgr2
set group2_licenses=foundation
:: Casual users
set group3_users=user1,user2,user3,user4,user5,user6,user7,user8,user9,user10,user11,user12,user13
set group3_licenses=foundation,found_aax

:: Determine license to execute if not specified as a command line option.
if not defined license (
  :: Add or remove from this list according to the defined groups.
  if defined group1_users call %currentpath%\setlicense.bat "%group1_users%" "%group1_licenses%"
  if defined group2_users call %currentpath%\setlicense.bat "%group2_users%" "%group2_licenses%"
  if defined group3_users call %currentpath%\setlicense.bat "%group3_users%" "%group3_licenses%"
)

:: Syntax: call mapdrive.bat %pro_system_drive% %pro_system_share% %pro_server% %pro_server_path%
:: Example: call mapdrive.bat P: ProSystem vserver01 D:\ptc\ProSystem
call mapdrive.bat %pro_system_drive% %pro_system_share% %pro_server% %pro_server_path%

:: Set the startup working directory
if not exist %proe_work_dir%\ (mkdir %proe_work_dir%)
cd /d %proe_work_dir%

:: Create the local cache folder if it doesn't exist
if not exist %ptc_wf_root% (mkdir %ptc_wf_root%)

:: Set the server registry file
call robocopy /mir %pro_system_drive%\configs\config.fld %ptc_wf_root%\Settings\config.fld

:: Add Company Standard Template files
robocopy /mir %pro_system_drive%\templates\ %proe_load_point%\templates2\

:: Add Company Standard Configuration files
robocopy %pro_system_drive%\configs\ %proe_load_point%\text\ /IF config.sup,config.win,config.pro,tree.cfg
```



```

:: Remove logs, and other temporary files
call %currentpath%\cleanworkdir.bat

:: Launch Pro/ENGINEER
echo "call %proe_load_point%\bin\proe.exe %proe_load_point%\bin%\license%.psf"
call %proe_load_point%\bin\proe.exe %proe_load_point%\bin%\license%.psf

```

proestart5.bat



Proestart5.bat

```

:: proestart5.bat
:: Configure the Pro/ENGINEER Wildfire 5.0 environment

:: Set the text editor command for use in Pro/ENGINEER
if exist C:\ptc\proeWildfire5.0\bin\proe.exe (set proe_drive=C)
if exist D:\ptc\proeWildfire5.0\bin\proe.exe (set proe_drive=D)
set proe_load_point=%proe_drive%\ptc\proeWildfire5.0

:: Set Environment Variables
set proe_work_dir=D:\workdir5.0\
:: Set the Windchill local cache location per the logged in user
set ptc_wf_root=%proe_work_dir%\%wcusername%\wf
:: Mapped Drive variables
set pro_system_drive=P:
set pro_system_share=ProSystem
set pro_server=mmeadows01
set pro_server_path=D:\ptc\ProSystem
:: Add Robocopy to the path for the local user.
set path= %pro_system_drive%\utilities;%PATH%
:: Set the UG load point
set ug_install_dir=C:\ug
if %computername%=="mmeadows01" (set ug_install_dir=D:\UG)
:: Set the text editor to use with Pro/ENGINEER
set text_editor=%ProgramFiles%\Windows NT\Accessories\wordpad.exe

:: Group users and licenses
:: Heavy users
set group1_users=heavy1,heavy2
set group1_licenses=flex3c
:: Manager users
set group2_users=mgr1
set group2_licenses=foundation
:: Casual users
set group3_users=user1,user2,user3
set group3_licenses=foundation,found_aax

:: Determine license to execute if not specified as a command line option.
if not defined license (
  :: Add or remove from this list according to the defined groups.
  If defined group1_users call %currentpath%\setlicense.bat "%group1_users%" "%group1_licenses%"
  If defined group2_users call %currentpath%\setlicense.bat "%group2_users%" "%group2_licenses%"
  If defined group3_users call %currentpath%\setlicense.bat "%group3_users%" "%group3_licenses%"
)

:: Syntax: call mapdrive.bat %pro_system_drive% %pro_system_share% %pro_server% %pro_server_path%
:: Example: call mapdrive.bat P: ProSystem vserver01 D:\ptc\ProSystem
call mapdrive.bat %pro_system_drive% %pro_system_share% %pro_server% %pro_server_path%

:: Set the startup working directory
if not exist %proe_work_dir%\ (mkdir %proe_work_dir%)
cd /d %proe_work_dir%

:: Create the local cache folder if it doesn't exist
if not exist %ptc_wf_root% (mkdir %ptc_wf_root%)

:: Set the server registry file

```

```

call robocopy /mir %pro_system_drive%\configs\config.fld %ptc_wf_root%\Settings\config.fld

:: Add Company Standard Template files
robocopy /mir %pro_system_drive%\templates\ %proe_load_point%\templates2\

:: Add Company Standard Configuration files
robocopy %pro_system_drive%\configs\ %proe_load_point%\text\ /IF config.sup,config.win,config.pro,tree.cfg

:: Remove logs, and other temporary files
call %currentpath%\cleanworkdir.bat

:: Launch Pro/ENGINEER
echo "call %proe_load_point%\bin\proe.exe %proe_load_point%\bin%\license%.psf"
call %proe_load_point%\bin\proe.exe %proe_load_point%\bin%\license%.psf

```

mapdrive.bat



Mapdrive.bat

```

:: mapdrive.bat
:: Map the given drive letter for the local user.
:: The script:
:: 1) Checks for the existence of the drive mapping
:: 2) Validates the drive mapping is the correct location for the script
:: 3) Removes and re-maps the drive if necessary.
:: This works on the server (subst) or from the end user's machine (net use).

:: Set Environment Variables
set map_drive_letter=%1
set shared_folder_name=%2
set server_name=%3
set server_local_path=%4

:: Decision Logic
:: 1) Is this the drive mapping correct?
:: A) No: Does the drive mapping exist?
:: I) Yes: Is it the server?
:: a) Yes: Remove drive substitution
:: b) No: Remove drive mapping
:: II) No: Map or substitute the drive
:: a) Is it the server?
:: i) Yes: substitute drive
:: ii) No: map drive

if not exist %map_drive_letter%\bin\proestart.bat (
  if exist %map_drive_letter% (
    if %server_name%==%computername% (
      subst %map_drive_letter% /d
    ) else (
      net use %map_drive_letter% /delete
    )
  )
  if %server_name%==%computername% (
    subst %map_drive_letter% %server_local_path%
  ) else (
    net use %map_drive_letter% \\%server_name%\%shared_folder_name% /PERSISTENT:Yes
  )
)
)

```

Some of the more mundane tasks like mapping a network drive and cleaning the startup working directory were encapsulated in standalone scripts: mapdrive.bat, clean.bat, cmdlineargs.bat.

clean.bat



Clean.bat

```
:: clean.bat
:: Cleans the working directory
setlocal

:: Purge the working directory
call purge.bat

:: Define the file names to delete
set
trash=.als,.bdm,.bom,.ers,.err,error.,errors.,iges_stats.dat,.inf,.info,.ger,.log,.lst,mechevnt,mmenglog,mmenglog.,memb,.m_p,.p
ls,.prd,s*.sec,.seq,std.,.tst
:: Note: '.crc' files should not be deleted. Circular references should be corrected.
:: Enable deletion of trail files 'trail.txt' if no one will ever run a trail file.
call :parse "%trash%"
goto :eos

:parse
:: Uses reduction to work through the list of servers.
set list=%1
set list=%list:="%=
FOR /f "tokens=1* delims=" %a IN ("%list%") DO (
  if not "%a" == "" call :sub %a
  if not "%b" == "" call :parse "%b"
)
goto :eos

:: Backup the last trail file
copy /y %proe_work_dir%\trail.txt* %proe_work_dir%\last_trail_backup.txt

:: Remove last trail file
del /Q /F %proe_work_dir%\trail.txt*

:sub
:: The meat of the script is here.
set _type=%1
if exist *%_type%* erase /q /q *%_type%*
goto :eos

:eos
endlocal
```

cmdlineargs.bat



Cmdlineargs.bat

```
:: cmdlineargs.bat
:: Process the command line arguments.

:: Users can specify command line arguments in the shortcut or at the command line.
:: Syntax: proestart.bat %1 %2 %3
:: Where each argument can be one of the following:
:: Pro/ENGINEER release number (e.g. release:3.0, release:5.0)
:: Windchill username (e.g. wcusername:mmeadows)
:: The username is not the computer login. It is the Windchill login name.
:: Pro/ENGINEER license name (e.g. license=foundation, license:found_adv, license:flex3c)
:: The license name should match the name of a *.psf file.
:: Example: proestart.bat release:5.0 username:user1

:: Iterate through the command line options and set the appropriate environment variables.
:: The number of options is not known but the maximum is 3.
```

```

call :sub %1
call :sub %2
call :sub %3

:sub
  set _pair=%1
  :: Trim the double quotes off the name-value pair
  set _pair=%_pair:~1,-1%
  :: Set the environment variable
  set %_pair%
goto :eos

:eos

```

setlicense.bat



Setlicense.bat

```

:: setlicense.bat
:: Syntax: call %currentpath%\setlicense.bat "%users%" "%licenses%"
:: Where: %users% and %licenses% are comma separated lists (no spaces) of users and license options respectively
:: This script can handle 1 to n of either users or licenses.
:: The double quotes around the variables are necessary.
:: This scrip compares the wusername to the users list and set the license options accordingly.

:: Set the environment variables
set count=1
set users=%1
set licenses=%2
:: Remove the double quotes from the command line variables
for /f "useback tokens=*" %%a in ("%users%") do set users=%%~a
for /f "useback tokens=*" %%a in ("%licenses%") do set licenses=%%~a

:: Parse the list of users
call :parse_users "%users%"
goto :eos

:parse_users
:: Uses reduction to work through the list of user.
set users=%1
set users=%users: "= %
FOR /f "tokens=1* delims=," %%a IN ("%users%") DO (
  if not "%%a" == "" call :sub_users %%a
  if not "%%b" == "" call :parse_users "%%b"
)
goto :eos

:sub_users
set _user=%1
:: Match wusername to the list of users.
if %wusername%==%_user% (call :parse_licenses "%licenses%")
goto :eos

:parse_licenses
:: Uses reduction to work through the list of licenses and build a list for prompting the user.
set licenses=%1
set licenses=%licenses: "= %
FOR /f "tokens=1* delims=," %%a IN ("%licenses%") DO (
  if not "%%b" == "" (
    if %count% == 1 (
      echo.
      echo.
      echo License Options:
    )
  )
  if not "%%a" == "" call :sub_licenses %%a

```

```

if not "%b" == "" (call :parse_licenses "%b")
if "%b" == "" (
    if %count% == 1 set license_name=%a
    if not %count% == 1 set /P license_name="What license would you like to run (e.g. %a)?"
)
)
set license=%license_name%
goto :eos

:sub_licenses
set _license=%1
:: For each license, add to the prompts.
echo %count%: %_license%
set /a count+=1
goto :eos

:eos

```

Another approach to license selection is demonstrated in setlicense.bat example script. Lists of users and available license options are specified in prostart3.bat and prostart5.bat which allows the Pro/ENGINEER Wildfire 5.0 testing team to call out specific licenses that may not be available to Pro/ENGINEER Wildfire 3.0 users. The license resolution logic is located in setlicense.bat.

This example is more complex than the previous examples. It compares a list of users to the Windchill username. If a match is found, the script determines what licenses are available to the user. If one license is available, it sets the license and launches Pro/ENGINEER. If multiple licenses are identified, it prompts the user to choose a license option by typing in its name at the prompt.

The licensing portion of this script can be expanded to check for available licenses and only offer the user a choice of available licenses.

Summary

Pro/ENGINEER launch scripts are a valuable tool for setting and enforcing corporate standards. They can be simple or complex depending on the needs of the users, the skills set of the author, and the time available for developing them.

The scripting language is not important as long as it supports the user environment and the author is comfortable using the language. Command line scripting was used in this document because it is simple to understand for non-programmers and universally supported on all Windows platform.

Pro/ENGINEER system administration is a continuous improvement process and the scripts described in this document have matured over many years through evolving user requirements. It is expected other Pro/ENGINEER requirements will surface and environment specific scripts will continue to evolve.

Use this documentation as a guide for developing environment specific scripts. Start with simple requirements like set the working directory, remove the temporary files and launch Pro/ENGINEER. Add to the script as needed.