Instrument update procedure to add and configure Whitelisting Software

This document details the steps required to add the McAfee whitelisting software to a system. The expected OS version is Windows 10 (and variants), but the procedure has not been tested on variants other than the Windows 10 IoT version.

The McAfee Application Control (MAC) whitelisting software is intended to augment the Windows anti-malware software included by default in the OS. It restricts any software from running on the instrument which was not already present when the MAC software was enabled.

The MAC software allows additional programs to be added manually, to handle late-added or optional components.

To allow applications to be updated, the MAC software allows components to be designated as updaters, with the ability to modify installed components or add new additional components.

Installation and configuration of the McAfee Application Control software involves multiple steps beyond the installation and configuration of the MAC software itself. These steps modify the system environment to allow instrument components to function in the controlled system environment, install and configure the Application Control software, designate applications and folders which contain updater applications, and identify contents or folders allowed to be modified during normal system operation. Some instructions will not be applicable for all instruments. Specifically, those instructions regarding the use of Windows PowerShell are specific to the instrument originally used with this configuration procedure. If the target system will have the command prompt accessible to the administrative user performing the installation and configuration, those instructions detailing the steps for adding the appropriate PowerShell Start Menu addition may be ignored.

MAC software may be configured to use certificates to authorize software, but this procedure does not use that method.

Note that if software is added or updated after the initial MAC configuration, the ‘solidification’ process must be repeated to allow detection and enumeration of any new expected software components and/or updaters.

When software is updated, the MAC software must be placed in the ‘update’ mode to avoid rejection of installations or configuration data. Failure to put the MAC software into the ‘update’ mode may result in disk image corruption requiring a complete re-image of the drive!

**MAC Configuration information**

The version of the McAfee Application Control software installed by the initial test of these instructions was version 8.2.143. Later versions may be used but have not been tested under this procedure. A specific MAC version will not be referenced in the instructions.

Installation will require the S/N and License key values. The tested version uses the following value as both S/N and license key:

0710-2208-1402-0108-2708

Installation on a fully configured instrument must be performed under an administrative user ID.

The update procedure has two major processes:

* to prepare the system to provide the required administrative privileges.
* to perform the software installation and configuration for the McAfee application Control software.

## System Preparation

The following steps may be omitted if the target OS configuration will allow administrative use of the command prompt. Go directly to the **McAfee Application Control Whitelisting Software Installation** section below.

Follow the steps below to prepare a fully configured system for the whitelisting application and the modification to the normal software update process.

1. Login under an administrative user ID.
2. Open File Explorer and navigate to the following folder:

C:\Users\<admin-user-name>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\WindowsPowerShell

1. Locate the “Windows PowerShell" shortcut and right-click on it.
2. Select “Copy”, then “Paste” the copy into the same folder.
3. Rename the copied shortcut from “Windows PowerShell - Copy” to “Windows PowerShell (Admin)”.
4. Right-click on the renamed shortcut and select “Properties” from the context menu.
5. On the “Properties” dialog, select the “Shortcut” tab if it is not already selected.
6. Click on the “Advanced” button, then select the “Run as Administrator” check-box on the “Advanced Properties” dialog. Click OK to accept and apply the change.
7. Click “Apply” then “OK” to dismiss the shortcut “Properties” dialog.
8. In File Explorer, right-click on the modified shortcut and select “Copy”. The modified shortcut will be pasted into all other administrative user start menus in the steps below.
9. Using File explorer, expand each administrative user under the “C:\Users” folder to reveal the following folder:

C:\Users\<admin-user-name>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\WindowsPowerShell

1. Right-click on the revealed “Windows PowerShell” folder and select “Paste” to copy the modified shortcut to the Start Menu for each admin user.
2. Ensure that the modified shortcut is pasted into the “Start Menu” folder structure for all administrative users. *Ensure that any customer-created administrative user accounts are updated as well.*

The following steps detail the modification of an instrument-specific custom software installation script used when the installer cannot be run from attached external drives. If the instrument application software may be installed directly from an external drive (e.g. USB drive), these steps may be skipped and you may go directly to the **McAfee Application Control Whitelisting Software Installation** section below.

1. Using File Explorer, navigate to the following folder:

C:\Users\<admin-user-name>\Desktop

1. Select the folder to display the desktop contents. Right click on the “Load <instrument or application name> Software” shortcut and select “Properties”.
2. On the “Properties” dialog, select the “Shortcut” tab if it is not already selected.
3. Click on the “Advanced” button, then select the “Run as Administrator” check-box on the “Advanced Properties” dialog. Click OK to accept and apply the change.
4. Click “Apply” then “OK” to dismiss the shortcut “Properties” dialog.
5. In File Explorer, right-click on the modified shortcut and select “Copy”. The modified shortcut will be pasted into all other administrative user desktops in the steps below.
6. Using File explorer, expand each administrative user under the “C:\Users” folder to reveal the following folder:

“C:\Users\<admin-user-name>\Desktop”

*Ensure that any customer-created administrative user accounts are updated as well.*

1. Right-click on the revealed “Desktop” folder and select “Paste” to copy the modified shortcut to the desktop of each admin user.

## McAfee Application Control Whitelisting Software Installation

This section describes the procedure for installing and configuring the McAfee Application Control software on a fully configured instrument OS. The installation and configuration steps are two different actions and may be performed separately or combined in a single operation. The target OS configuration may not yet include instrument application software. However, installation on early development or prototype OS versions may require several iterations of the configuration steps as the content of the configured OS changes.

The ***installer*** may be run from a command prompt window or File Explorer. Use of File Explorer to start the installer is not recommended, as it may limit interaction and may close the installer window without allowing the contents of the window to be captured or examined. Also, when using File Explorer, the installation and configuration steps must be performed separately.

When using a command window, either the standard ‘cmd.exe’ or a ‘PowerShell’ command window may be used, depending on availability in the target instrument environment. All run environments (command window or File Explorer) require the ‘Run as administrator’ property to be set.

The ***configuration*** of the MAC software requires user interaction and must be performed in a command prompt window. This may require the administrative PowerShell shortcuts discussed and created earlier in this document.

The following procedure will perform ***both the installation and configuration*** of the MAC software in a command window using a batch script which will run the installation and the configuration steps with minimal user interaction. Installation instructions will not reference the specific type of command window. Confirmation steps in the installer script will ensure that the installation steps are completing correctly. An example of the installation-configuration script is provided at the end of this document. The example uses configuration information for the V-CELL instrument, and will require modification for use in other target instruments.

As previously noted, all software must be installed under an administrative user id. Running under an administrative user session is not sufficient to ensure all required privileges.

These instructions will assume the installer is in a “McAfee\Solidcor821-143\_Win” sub-folder on an external USB drive (ex: H:\McAfee\Solidcor821-143\_Win). If using other versions, ensure the folder structure matches the “McAfee\Solidcor***xyz***” pattern.

1. Locate the version of the McAfee software to be used and copy it to a USB drive. Ensure the “Solidcor***xyz***” folder is located below the McAfee folder on the USB drive.
2. The installer folder (ex: “H:\McAfee\Solidcor821-143\_Win”) should contain the installation script file “InstallWhitelisting.bat”. If the installation script is not present in the folder, copy it to the installer folder on the instrument using File Explorer.
3. If not using an instrument-specific application software installer script, go to step 6.
4. Locate the application instrument installer-loader script file.
5. Copy the installer-loader script file to the McAfee folder on the USB drive (ex: H:\McAfee\<application-name>\_Installer.bat).

NOTE: installer name format is not specific; substitute the appropriate name.

1. Plug the USB drive with the application software into a USB port on the instrument.
2. Copy the complete application software installer folder to a temporary folder on the instrument drive. If the “temp” folder on the instrument drive is used, the resulting path should be “C:\temp\McAfee\Solidcor821-143\_Win”.
3. Start a command window. Remember that the command window must be started with the ‘Run as Administrator’ property set.
4. In the command window, navigate to the temporary folder now containing the application software installer by typing “cd <installer\_folder>” (ex: cd "C:\temp\McAfee\Solidcor821-143\_Win"). Remember to quote the installer folder path to ensure correct interpretation by the command handler if the path contains embedded spaces.
5. In the command window, type .\InstallWhitelisting.bat to start the installation and configuration script. The leading “.\” is required.
6. The batch script may display some of the following confirmation and interaction steps in different order than presented here. Use the steps below to ensure correct entries where required.

* If prompted, accept the terms of the “End User License Agreement” and continue.
* If prompted, accept the default installation location.
* If prompted for User Name, Organization/Company, and Serial Number information, enter the following:
  + User Name: Beckman Coulter
  + Organization/Company: Beckman Coulter
  + Serial Number: 0710-2208-1402-0108-2708 (or the appropriate current license value)
* If prompted to create a desktop shortcut, ensure the option is deselected.

1. If the ***installation*** returns an error, the installer script will display an error message and wait for user confirmation. Pay close attention to the installation window for indications of errors. An error during the installer script processing does not close the command window.
2. After successful installation, the script will begin the ***configuration*** steps, starting with application features. After configuration of the application features, the installer script will display a list of features, followed by the following confirmation prompt:
   * + “Check the list of features. Does it match the expected? (y,n):”
3. Verify the displayed list matches the following list:
   * + checksum Enabled
     + deny-read Disabled
     + deny-write Disabled
     + discover-updaters Enabled
     + execution-control Enabled
     + integrity Enabled
     + mp Enabled
     + mp-casp Enabled
     + mp-nx Enabled
     + mp-vasr Enabled
     + mp-vasr-forced-relocation Enabled
     + network-tracking Enabled
     + pkg-ctrl Disabled
     + script-auth Enabled
4. If discrepancies are noted, enter ‘N’ at the confirmation prompt to exit the installer script. If the displayed list matches the expected list, enter ‘Y’ to continue.
5. After successful configuration, the script will continue the configuration steps, configuring the list of trusted folder locations. After configuration of the trusted folder locations, the installer script will display a list of trusted folders, followed by the following confirmation prompt:
   * + “Check the list of trusted folders. Does it match the expected? (y,n):”
6. Verify the displayed list matches the expected list. The list will be determined by the target instrument and requires the install-configuration script to be modified. The list example below is from the ViCell instrument.
   * + + updater “C:\Instrument”
     + + “C:\Instrument\bin”
     + + “C:\Instrument\Config”
     + + “C:\Instrument\DB\_Data”
     + + “C:\Instrument\Export”
     + + “C:\Instrument\ExternalImages”
     + + “C:\Instrument\Logs”
     + + “C:\Instrument\ResultsData”
     + + “C:\Instrument\Software”
     + + “C:\Instrument\SW\_Install”
     + + “C:\Instrument\Tools”
     + + “C:\Instrument\Tools\.NET\_4.8”
     + + “C:\Instrument\Tools\Basler”
     + **+ “C:\Instrument\Tools\BeckmanConnect”**
     + + “C:\Instrument\Tools\EDB”
     + + “C:\Instrument\Tools\fciv”
     + + “C:\Instrument\Tools\Omicron”
     + + “C:\Instrument\Tools\vc”
     + **+ "C:\Program Files (x86)\TeamViewer"**
     + **+ "C:\ProgramData\Beckman Coulter\Connect\bin"**
     + + ”C:\Program Files\PostgreSQL\10\bin”
     + + "C:\Program Files\PostgreSQL\10\lib"
     + + "C:\Program Files\PostgreSQL\10\scripts"
     + + "C:\Program Files\PostgreSQL\10\share"
     + + "C:\Program Files\PostgreSQL\10\pgAdmin 4"

NOTE: if the target instrument will support the ***optional*** installation of the BeckmanConnect application, the three highlighted lines of the example must be present in the configuration list even if not present on the system at the time of configuration.

NOTE: as shown above, remember to quote folder paths to ensure proper handling of paths containing embedded spaces.

1. If discrepancies are noted, enter ‘N’ at the confirmation prompt to exit the installer script. If the displayed list matches the expected list, enter ‘Y’ to continue.
2. After successful configuration, the script will continue the configuration steps, configuring the list of trusted updater applications and scripts. After configuration of the updaters, the installer script will display a list of all updater processes configured by both the application installation and the secondary installer script. The installer script displays the following confirmation prompt:
   * + “Check the list of trusted update processes. Does it match the expected? (y,n):”
3. The displayed list will contain applications and components the Application Control software has predetermined to be updaters, as well as the values added by the installer script. The list is long, but normally displayed in alphabetical order. Verify the displayed list contains all the script-added items specific to the target instrument and added to the installer-script configuration list. The line elements for the list entries may differ in the content of the initial elements (the “AUTO\_X” element), but the final element on the lines should match the formats of those items added to the configuration list in the script. The list example below is from the ViCell instrument showing the expected updater list with expected locations. Note that not all the listed names will exist on the system and can contain anticipated variants.
   * + -t AUTO\_2 \Instrument\InstallViCell.bat
     + -t AUTO\_3 \Instrument\InstallViCellBLU.bat
     + -t AUTO\_4 \Instrument\InstallViCellFL.bat
     + -t AUTO\_5 \Instrument\LoadViCell\_Installer.bat
     + -t AUTO\_6 \Instrument\ViCell\_Install.exe
     + -t AUTO\_7 \Instrument\ViCell\_Install.exe -p \Instrument\LoadViCell\_Installer.bat
     + -t AUTO\_8 **bcupdate.exe**
     + -t AUTO\_9 **BeckmanConnectSetup.exe**
     + -t AUTO\_10 ViCellBLU\_UI.msi
     + -t AUTO\_11 ViCellFL\_UI.msi
4. If discrepancies are noted, enter ‘N’ at the confirmation prompt to exit the installer script. If the displayed list matches the expected list, enter ‘Y’ to continue.
5. After successful configuration, the script will continue the configuration steps, configuring the list of updater-permitted certificates installed. After configuration of the certificates, the installer script will display a list of all the updater-permitted certificates. The installer script displays the following confirmation prompt:
   * + “Check the list of installed updater-permitted certificates. Does it match the expected? (y,n):”
6. The displayed list will contain applications and components the Application Control software has determined are installed on the system, as well as the values added by the installer script. The list may be long, but normally displayed in alphabetical order. Verify the displayed list contains all the script-added items specific to the target instrument and added to the installer-script configuration list. The line elements for the list entries may differ in the content of the initial elements (the “AUTO\_X” element), but the final element on the lines should match the format and content of those items added to the configuration list in the script. The list example below is from the ViCell instrument showing the expected updater-permitted certificate list.
   * + AUTO\_15 681c5a009d210131b0329ffbf95bc062c4f7380f1edf70e84d32ccad1b4fab5a
     + AUTO\_16 5b7582701ef966127c7b828b225f082c08a4e610d2af4ed20d7ab49cd7261a6d
7. After successful configuration, the script will continue the configuration steps, examining the system for OS components, drivers, and additional installed applications. The examination process takes from 20 to 50 minutes to complete. If errors are encountered, the script will display an error message and wait for user confirmation before exiting. Again, note that exiting the script does not close the command window.
8. After successfully examining the system and compiling the list of components, the installer script will configure the MAC software to start when the system starts and will perform an instrument computer restart.
9. After the computer restarts, confirm that no errors are encountered during the instrument startup, or Vi-CELL application startup.
10. After completion of the installation, remove all installation files in the “C:\temp” folder (or wherever the files were placed).

## McAfee Application Control Whitelisting Software Installation

The example script shows the content of a MAC installation-configuration script from a Vi-CELL instrument. All Vi-CELL folder and application references would need to be updated to use those from the target instrument. Pay particular attention to the ensuring the list of trusted applications and trusted folders is appropriate and complete for the target instrument.

Of note are the commands used to control the MAC software for the enable, disable, update, and end of update conditions. The key commands are listed below, and can be found in the installer-configuration script example.

sadmin enable : puts the MAC software into the “enable on restart” mode; computer restart is required for completion

sadmin disable : disables the MAC software; no restart is necessary, but good practice

sadmin begin-update : puts the MAC software into the update mode allowing components to be updated or changed;

sadmin end-update : closes the update session; no restart required

: not required but helpful

sadmin status : shows the enable/disable status of the MAC software

sadmin help : display the top-level commands and syntax, including accessing help for individual parameters commands;

The example script is presented on the following pages.

#########################################################################

@echo off

set FEATURES\_OK=

set TRUSTED\_OK=

set UPDATERS\_OK=

set CERTS\_OK=

set DEBUG\_MODE=

set DISPLAY\_TIMEOUT=

set APP\_INSTALLER\_LOADER=

if "%1x" == "x" goto start

if "%1x" == "-Dx" goto set\_debug

if "%1x" == "-dx" goto set\_debug

@echo Unrecognized parameter "%1". Ignored...

@echo.

goto start

:set\_debug

set DEBUG\_MODE=YES

shift

@echo on

:start

If “%APP\_INSTALLER\_LOADER%x” == “x” set APP\_INSTALLER\_LOADER=LoadViCell\_Installer.bat

pushd .

if exist \ViCell-OS-v0.10f.txt goto start\_ok

if exist \ViCell-OS-v0.11a.txt goto start\_ok

if exist \ViCell-OS-v0.11b\*.txt goto not\_required

if exist \ViCell-OS-v0.11c\*.txt goto not\_required

if exist \ViCell-OS-v1.\*.txt goto not\_required

@echo.

@echo Error installing the McAfee Application Control software. Base OS version not supported. Script will exit in 60 seconds.

goto err\_xit3

:not\_required

@echo.

@echo Base OS version does not require McAfee Application Control. Script will exit in 60 seconds.

goto err\_xit3

:start\_ok

rem running the installer after it is already installed will REMOVE the application if not yet enabled,

rem or it will fail because the application will block removal of itself when it is enabled and active.

if exist "C:\Program Files\McAfee\Solidcore\sadmin.exe" goto begin\_config

rem NOTE: can't use system environment variables, as the embedded backslash characters aren't 'escaped' for use in embedded strings

**rem Ensure the following command is typed as a complete line, not broken into two lines as may be created and displayed by the Notepad text wrapping function!**

**rem Enter the following line EXACTLY as shown, including backslash escaping sequences and double quoting of the entire /v parameter string including embedded strings!**

CALL .\setup-win-8-10-2012-amd64-8.2.1.143.exe /v" SERIALNUMBER=0710-2208-1402-0108-2708 SHORTCUT=0 INSTALLDIR=\"C:\\Program Files\\McAfee\\Solidcore\" /l+\*v \"C:\\temp\\McAfee\\S3Setup.log\""

if errorlevel == 0 goto begin\_config

goto err\_xit1

:begin\_config

if not exist "C:\temp\McAfee\%APP\_INSTALLER\_LOADER%" goto chk\_loader

copy /y /v "C:\temp\McAfee\%APP\_INSTALLER\_LOADER%" "C:\Instrument"

goto config\_features

:chk\_loader

if exist "C:\Instrument\LoadViCell\_Installer.bat" goto config\_loader

@echo.

@echo Error installing the McAfee Application Control software. Missing installer-loader in base OS. Script will exit in 60 seconds.

goto err\_xit3

:config\_loader

echo sadmin begin-update > C:\Instrument\sadmin\_begin.bat

echo sadmin end-update >> C:\Instrument\%APP\_INSTALLER\_LOADER%

copy /a /v /y C:\Instrument\sadmin\_begin.bat + C:\Instrument\%APP\_INSTALLER\_LOADER% C:\Instrument\%APP\_INSTALLER\_LOADER%

if exist C:\Instrument\sadmin\_begin.bat del /f /q C:\Instrument\sadmin\_begin.bat

:config\_features

cd "C:\Program Files\McAfee\SolidCore"

sadmin features disable deny-write

sadmin features disable pkg-ctrl

@echo.

sadmin features

:chk\_features

@echo.

@set /p FEATURES\_OK=Check the list of features. Does it match the expected? (y,n):

if "%FEATURES\_OK%x" == "Yx" goto config\_trusted

if "%FEATURES\_OK%x" == "yx" goto config\_trusted

if "%FEATURES\_OK%x" == "Nx" goto err\_xit2

if "%FEATURES\_OK%x" == "nx" goto err\_xit2

@echo.

@echo Unrecognized entry!

goto chk\_features

:config\_trusted

sadmin trusted -i -u C:\Instrument

sadmin trusted -i C:\Instrument\bin

sadmin trusted -i C:\Instrument\Config

sadmin trusted -i C:\Instrument\DB\_Data

sadmin trusted -i C:\Instrument\Export

sadmin trusted -i C:\Instrument\ExternalImages

sadmin trusted -i C:\Instrument\Logs

sadmin trusted -i C:\Instrument\ResultsData

sadmin trusted -i C:\Instrument\Software

sadmin trusted -i C:\Instrument\SW\_Install

sadmin trusted -i C:\Instrument\Tools

sadmin trusted -i C:\Instrument\Tools\.NET\_4.8

sadmin trusted -i C:\Instrument\Tools\Basler

sadmin trusted -i C:\Instrument\Tools\BeckmanConnect

sadmin trusted -i C:\Instrument\Tools\EDB

sadmin trusted -i C:\Instrument\Tools\fciv

sadmin trusted -i C:\Instrument\Tools\Omicron

sadmin trusted -i C:\Instrument\Tools\vc

sadmin trusted -i "C:\Program Files (x86)\TeamViewer"

sadmin trusted -i "C:\ProgramData\Beckman Coulter\Connect\bin"

sadmin trusted -i "C:\Program Files\PostgreSQL\10\bin"

sadmin trusted -i "C:\Program Files\PostgreSQL\10\lib"

sadmin trusted -i "C:\Program Files\PostgreSQL\10\scripts"

sadmin trusted -i "C:\Program Files\PostgreSQL\10\share"

sadmin trusted -i "C:\Program Files\PostgreSQL\10\pgAdmin 4"

@echo.

sadmin trusted -l

:chk\_trusted

@echo.

@set /p TRUSTED\_OK=Check the list of trusted folders. Does it match the expected? (y,n):

if "%TRUSTED\_OK%x" == "Yx" goto config\_updaters

if "%TRUSTED\_OK%x" == "yx" goto config\_updaters

if "%TRUSTED\_OK%x" == "Nx" goto err\_xit2

if "%TRUSTED\_OK%x" == "nx" goto err\_xit2

@echo.

@echo Unrecognized entry!

goto chk\_trusted

:config\_updaters

sadmin updaters add \Instrument\InstallViCell.bat

sadmin updaters add \Instrument\InstallViCellBLU.bat

sadmin updaters add \Instrument\InstallViCellFL.bat

sadmin updaters add \Instrument\LoadViCell\_Installer.bat

sadmin updaters add \Instrument\ViCell\_Install.exe

sadmin updaters add -p \Instrument\LoadViCell\_Installer.bat \Instrument\ViCell\_Install.exe

sadmin updaters add BeckmanConnectSetup.exe

sadmin updaters add bcupdate.exe

sadmin updaters add ViCellBLU\_UI.msi

sadmin updaters add ViCellFL\_UI.msi

@echo.

sadmin updaters list

:chk\_updaters

@echo.

@set /p UPDATERS\_OK=Check the list of trusted update processes. Does it match the expected? (y,n):

if "%UPDATERS\_OK%x" == "Yx" goto config\_certs

if "%UPDATERS\_OK%x" == "yx" goto config\_certs

if "%UPDATERS\_OK%x" == "Nx" goto err\_xit2

if "%UPDATERS\_OK%x" == "nx" goto err\_xit2

@echo.

@echo Unrecognized entry!

goto chk\_updaters

:config\_certs

rem update to add BeckmanConnect certificate; also as an updater

**rem NOTE that if the file is in the current folder, the file specifier REQUIRES the '.\' path designation!**

sadmin cert add -u ".\681c5a009d210131b0329ffbf95bc062c4f7380f1edf70e84d32ccad1b4fab5a.cer"

rem update to add TeamViewer certificate; also as an updater

sadmin cert add -u ".\5b7582701ef966127c7b828b225f082c08a4e610d2af4ed20d7ab49cd7261a6d.cer"

@echo.

sadmin cert list -u

:chk\_certs

@echo.

@set /p CERTS\_OK=Check the list of installed updater-permitted certificates. Does it match the expected? (y,n):

if "%CERTS\_OK%x" == "Yx" goto do\_solidify

if "%CERTS\_OK%x" == "yx" goto do\_solidify

if "%CERTS\_OK%x" == "Nx" goto err\_xit2

if "%CERTS\_OK%x" == "nx" goto err\_xit2

@echo.

@echo Unrecognized entry!

goto chk\_certs

:do\_solidify

sadmin so

if errorlevel == 0 goto do\_enable

goto err\_xit2

:do\_enable

sadmin enable

if exist \Instrument\ViCell-OS-\*.txt del /f /q \Instrument\ViCell-OS-\*.txt

echo ViCell-OS-v0.11c2 > \Instrument\ViCell-OS-v0.11c2.txt

if exist \ViCell-OS-\*.txt del /f /q \ViCell-OS-\*.txt

copy /y /v \Instrument\ViCell-OS-v0.11c2.txt \ViCell-OS-v0.11c2.txt

if not "%DEBUG\_MODE%x" == "YESx" goto do\_restart

@echo.

pause

:do\_restart

@echo.

@echo System must be restarted for all changes to take effect. The updater will restart the system in 60 seconds.

shutdown /r /t 60 /d u:4:2

rem shouldn't ever get here, but allow exit

goto xit2

:err\_xit1

@echo.

@echo Error installing the McAfee Application Control software. Script will exit in 60 seconds.

goto err\_xit3

:err\_xit2

@echo.

@echo Error configuring the McAfee Application Control software. Script will exit in 60 seconds.

:err\_xit3

set DISPLAY\_TIMEOUT=60

:xit

@echo.

if "%DISPLAY\_TIMEOUT%x" == "x" goto xit2

timeout /T %DISPLAY\_TIMEOUT%

:xit2

set FEATURES\_OK=

set TRUSTED\_OK=

set UPDATERS\_OK=

set DEBUG\_MODE=

set DISPLAY\_TIMEOUT=

set APP\_INSTALLER\_LOADER=

popd