

## Application Note #1179: Controlling other PC Programs from WinGP

### Introduction

There are a number of ways that a WinGP application can interface with other software running on the same IPC. In this application note we will concentrate on a feature which the manual calls "Running an Application from WinGP" or "Start Application".

This feature allows a program (typically an .EXE file) to be triggered via WinGP and to pass a parameter to it. A simple example would be to start Microsoft Excel via it's .EXE file and to pass the name of an Excel spreadsheet as it's parameter: i.e.

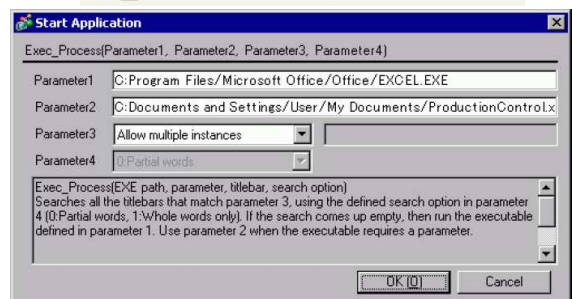
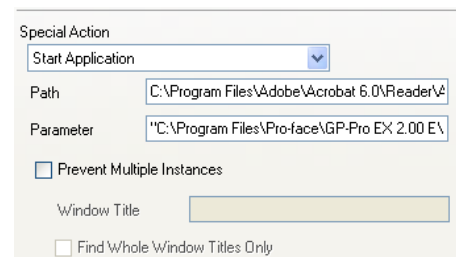
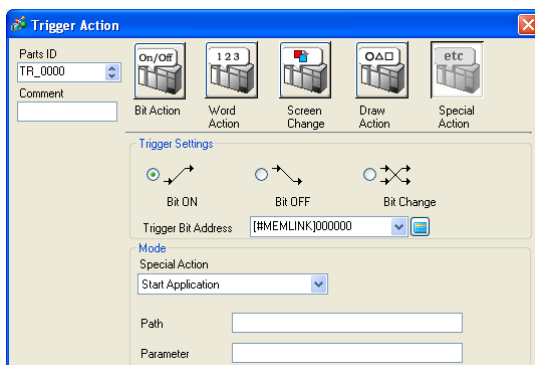
Program path : C:\Program Files\Microsoft Office\OFFICE11\Excel.exe  
Parameter : "C:\Documents and Settings\MyIPC\My Documents\MyFile.xls"

This function can be implemented from a Special Function Switch, a D-Script or a Trigger Action in WinGP.

The purpose of this application note is to show some of the other ways that this feature can be used to control other Windows programs, including starting and stopping, positioning and re-sizing etc without having to be a proficient PC programmer.

### The Basic "Start Application" function:

This can be run from Special Switch, D-Script or Trigger action programmed in the GPPRO-EX editor.



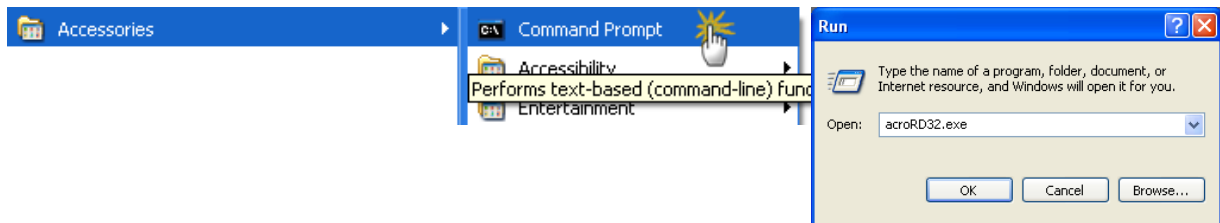
The parameters which can be set are the same for all three methods:

- Program Path and filename for the command
- Parameter string to pass to program. Use "" in case the string contains spaces.
- Option to prevent multiple instances of the specified Program Window.

### Command Prompt & Command Line Interface (CLI):

The engine behind the “Start Application” option is the Windows Command Prompt. Some users may recognize this as the direct successor to the old “DOS prompt” but it remains an integral part of the modern Windows environment (including Vista) and is fully supported by Microsoft. The Command Prompt is a program itself (CMD.EXE).

Many standard Windows applications can be started from the DOS prompt or via the “Run” command



The Command Line library in Windows includes a number of useful file and task commands including for example Copying or Renaming files, and ending a Program Task (taskkill).

Microsoft Command Line Reference A-Z:

<http://technet.microsoft.com/en-us/library/bb491071.aspx>

### Batch Files (\*.BAT):

Command line functions can be grouped together in Batch files which are simply text files with the suffix .BAT. Batch files can be used to perform multiple commands in sequences and can even perform program loops.

A .BAT file can be used as a program file in the Start Application function of WinGP. Apart from allowing multiple commands to be triggered from one switch, this also means that the effective action can be programmed outside the GPPRO-EX environment.

## Command Line options for Common Applications:

### Adobe Acrobat:

Task	Command Line
Open a PDF file:	AcroRd32.exe <i>PdfFile</i>
Open a PDF file in a new instance of Adobe Reader:	AcroRd32.exe /N <i>PdfFile</i>
Open a PDF file at page 7:	AcroRd32.exe /A "page=7=OpenActions" <i>PdfFile</i>
Open a PDF file with navigation pane active, zoom out to 50%, and search for and highlight the word "batch":	AcroRd32.exe /A "zoom=50&navpanes=1=OpenActions&search=batch" <i>PdfFile</i>
Print a PDF file with dialog:	AcroRd32.exe /P <i>PdfFile</i>
Print a PDF file silently:	AcroRd32.exe /N /T <i>PdfFile PrinterName</i> [ <i>PrinterDriver</i> [ <i>PrinterPort</i> ] ]

<http://partners.adobe.com/public/developer/en/acrobat/PDFOpenParameters.pdf#page=5>

### Internet Explorer

Task	Command Line
Open Internet Explorer without the home page:	iexplore.exe -nohome
Open Internet Explorer in Windows Explorer mode:	iexplore.exe -e <i>Folder</i>
Open Internet Explorer in kiosk mode:	iexplore.exe -k [ <i>url</i> ]

<http://www.microsoft.com/technet/prodtechnol/ie/reskit/6/part7/z03ie6rk.mspx?mfr=true>

### Powerpoint Viewer

Task	Command Line
Open a PowerPoint file in Slideshow mode:	pptview.exe <i>PowerPointFile</i>
Open a print dialog (without "splash screen"):	pptview.exe /S /P <i>PowerPointFile</i>

<http://support.microsoft.com/kb/830040>

## Windows Media Player

Task	Command Line
Play a file (audio or video)	Wmplayer.exe "c:\nameoffile.wma"
Open wmp and play the playlist listed	Wmplayer.exe /Playlist NameOfPlayList
Play a song or video in fullscreen mode:	Wmplayer.exe "c:\nameofvideo.wmv" /fullscreen

<http://support.microsoft.com/KB/241422>

## Excel:

Task	Command Line
Open an Excel spreadsheet:	excel.exe <i>ExcelSpreadsheet</i>
Open an Excel spreadsheet in read-only mode:	excel.exe /r <i>ExcelSpreadsheet</i>
Open an Excel spreadsheet in safe mode:	excel.exe /s <i>ExcelSpreadsheet</i>

<http://support.microsoft.com/kb/291288>

## Passing parameters to custom programs such as VB.NET:

Most PC programmers using Visual Basic or one of the other development environments can access data from a WinGP application using it's built-in API (Application Program Interface). This is the most effective way for VB programmers to interface to WinGP since it is direct and data can be passed in either direction.

However these environments also can also build .EXE files which can receive Parameters (Arguments) in the same way as most other Windows programs.

In the example shown on the right from vbdotnetheaven.com, two parameters are passed to the EXE as a string separated by Space " ". i.e.

proc.Start(Application.StartupPath & "\Sambandh.exe", "1 M")

<http://www.vbdotnetheaven.com>

```
Imports System.Windows.Forms
Module Module1
    Public Sub main()
        Dim frm As New Form1
        Dim str() As String
        str = GetCommandLineArgs()
        If (UBound(str) >= 0) Then
            a = str(0).ToString
            b = str(1).ToString
        End If
        Application.Run(frm)
    End Sub

    Function GetCommandLineArgs() As String()
        ' Declare variables.
        Dim separators As String = " "
        Dim commands As String =
            Microsoft.VisualBasic.Command()
        Dim args() As String =
            commands.Split(separators.ToCharArray)
        Return args
    End Function
End Module
```

**Appendix: Reference sources and further reading**

<http://www.robvanderwoude.com/commandlineswitches.php>

<http://www.vbdotnetheaven.com/UploadFile/pragati2478/HowtoPassParameterinEXE04052006084011AM/HowtoPassParameterinEXE.aspx>

<http://technet.microsoft.com/en-us/library/bb491071.aspx>

-----  
For more information on Pro-face and our full line of HMI, Operator Interface and Industrial PC products please visit our web site at [www.profaceamerica.com](http://www.profaceamerica.com).

For technical support email: [support@profaceamerica.com](mailto:support@profaceamerica.com) or call: 800.289.9266.

-----

© 2014 Pro-face America. Specifications may change without notice. Pro-face is a registered trademark of Digital Corporation. Other brand or product names are the property of their respective owners.