How to run a single instance of an application (part 1 of 4)

Contents, abstract and overview

Contents

- *Abstract*
- *Overiew*
- *An initial approach*
  - *Finding if your program is running*
  - *Passing the command line to a previous instance*
  - *Activating the Previous Instance*
  - *Finishing touches*
  - *Putting it all together*
- *An object oriented solution*
  - *Overview*
  - *The USingleInst unit*
  - *Modifications to the project file*
  - *Modifications to the main form*
  - *Overriding TSingleInst*
- *Demonstration code*
- *Conclusion*
- *Feedback*

Abstract

This article discusses how to ensure that just a single instance of an application can be run. An application does this by checking if an instance is already running and terminating it if so. We also look at how a duplicate instance can pass its command line parameters to the existing instance before terminating.

Overview

The method we will use to prevent multiple application instances is based on the detection of an application with a known main window class. Once such an application is detected we then pass any command line data via the *WM_COPYDATA* message.

Alternatives

Mutexes can be used to detect muliple application instances and memory mapped files can be used to exchange data. These techniques are not persued here.

Listing 1 outlines the methodology we will be using:

```
search for a top level window of the correct class
if such a window exists
  activate the window
```

```
  pass any command line data to the window using WM_COPYDATA
  terminate this application instance
else
  start this application as normal
  process the command line parameters
end
```

Listing 1

We will discuss an initial approach to this problem in the *next part*.