Communicate between processes using windows messaging

Windows messaging is an easy way for processes to communicate. Below, you will find an example of sending such messages, as well as an example message handler to receive them.

Please keep in mind that there is no guarantee that your process will receive this message. Or, it may receive it instantaneously. I recommend sending some sort of ACK response back to the originating app so I knows it was received properly.

Sender

```
// wmCopyData
// Allows inter-process communications via Windows WM_COPYDATA messaging.
procedure wmCopyData(WndClass:PChar;WndTitle:PChar;Msg:String);
var
  hWnd  : THandle         // Handle to target window to receive message
  cds   : CopyDataStruct; // Structure to package the outbound message
begin
  // Find target window
  hWnd := FindWindow(PChar(WndClass), PChar(WndTitle));
  try
    cds.dwData := 0
    cds.cbData := Length(Msg);   // Length of message
    cds.lpData := PChar(Msg);    // Actual message
    // The following function is not necessary for this to work
    SetForegroundWindow(hWnd);   // Pulls target window up top
    SendMessage(hWnd, wm_CopyData, 0, Integer(@cds));
send the message
  finally
    CloseHandle(hWnd)            // Close handle to target
  end;
end;
```

A typical call to this procedure would be:

```
wmCopyData('NOTEPAD','Untitled - Notepad','Test Message');
```

Keep in mind that *Notepad* does not have a handler to receive messages sent via *WM_COPYDATA*.

Also, it is not necessary to use both the Class name and Window title text, one is sufficient. However, if you have multiple copies open, it will go to the first one it finds.

Receiver

```
// Add a custom message handler so our app gets notified upon receipt
private
  procedure WMGetData(var Msg: TWMCopyData); message WM_COPYDATA;

// wmGetData
// Receives inbound messages - Callback function
// Called from message handler
procedure TForm1.wmGetData(var Msg: TWMCopyData);
var
  sText: array[0..255] of Char;  // Create an array to store message in
begin
  // Cast inbound data structure into a character array
  StrLCopy(sText, Msg.CopyDataStruct.lpData, Msg.CopyDataStruct.cbData);
  Edit1.Text := sText;
end;
```

Author:	Dennis LV
Contributor:	Dennis LV

Added:	2007-08-31
Last updated:	2007-08-31