

Execute a program and wait until it is done

The following function starts an executable with a given command line. If the execution fails (non-0 return value from executable), the return value is *False*, and *GetLastError* - \$2000 is the return value from the executable. If the execution is successful (return value 0 from the executable), the return value is *True*.

```
function ExecAndWait(sExe, sCommandLine: string): Boolean;
var
  dwExitCode: DWORD;
  tpiProcess: TProcessInformation;
  tsiStartup: TStartupInfo;
begin
  Result := False;
  FillChar(tsiStartup, SizeOf(TStartupInfo), 0);
  tsiStartup.cb := SizeOf(TStartupInfo);
  if CreateProcess(PChar(fsExe), PChar(sCommandLine), nil, nil, False, 0,
    nil, nil, tsiStartup, tpi) then
  begin
    if WAIT_OBJECT_0 = WaitForSingleObject(tpiProcess.hProcess, INFINITE) then
    begin
      if GetExitCodeProcess(tpiProcess.hProcess, dwExitCode) then
      begin
        if dwExitCode = 0 then
          Result := True
        else
          SetLastError(dwExitCode + $2000);
        end;
      end;
    end;
    dwExitCode := GetLastError;
    CloseHandle(tpiProcess.hProcess);
    CloseHandle(tpiProcess.hThread);
    SetLastError(dwExitCode);
  end;
end;
```

You can find a *similar routine* in the *Code Snippets Database*. For more control over the executed program see the *Console Application Runner Classes*.

Author:	Unknown
Added:	2007-06-02
Last updated:	2013-10-12

Copyright © Peter Johnson (DelphiDabbler) 2002-2018