Daemon oder Windows System Service

Daemon oder auch Service Routinen sind mit Free Pascal siehe wiki.freepascal.org/Daemons_and_Services (https://wiki.freepascal.org/Daemons_and_Services) leicht zu realisieren, hier ein kleines Beispiel inklusive EventLog-Ausgabe.

Eine Variante in der FreeOnTerminate des Threads auf true gesetzt ist, wodurch im Daemon.Stop einfach mit Terminate das beenden des Threads eingeleitet werden kann und mit WaitFor auf das Ende gewartet werden kann.

thedaemon.lpr (/code/thedaemon.lpr?mode=download) Pascal (5,42 kByte) 01.12.2016 20:29

```
// ******************************************************************************
//  Title..... :  The Daemon Example
//
//  Modulname ..... :  thedaemon.lpr (project file)
//  Type ..... :  Unit
//  Author ..... :  Udo Schmal
//  Development Status  :  01.12.2016
//  Operating System .. :  Win32/Win64
//  IDE ..... :  Lazarus
// ******************************************************************************
program thedaemon;

{$mode objfpc}{$H+}

uses
  HeapTrc,
{$IFDEF UNIX}{$IFDEF UseCThreads}
  CThreads,
{$ENDIF} Cmem,{$ENDIF}
  Classes, SysUtils, EventLog, DaemonApp;

type
  TTheThread = class(TThread)
    procedure Execute; override;
    destructor Destroy; override;
  end;

  TTheDaemon = class(TCustomDaemon)
  private
    FThread: TTheThread;
  public
    function Install: boolean; override;
    function UnInstall: boolean; override;
    function Start: boolean; override;
    function Stop: boolean; override;
    function Pause: boolean; override;
    function Continue: boolean; override;
    function Execute: boolean; override;
    function ShutDown: boolean; override;
  end;

  TTheDaemonMapper = class(TCustomDaemonMapper)
  public
    constructor Create(AOwner: TComponent); override;
    procedure ToDoOnInstall(Sender: TObject);
    procedure ToDoOnRun(Sender: TObject);
    procedure ToDoOnUninstall(Sender: TObject);
    procedure ToDoOnDestroy(Sender: TObject);
  end;

function BoolToStr(AVal: Boolean): String;
begin
  if AVal = True then result := 'true' else result := 'false';
end;

procedure TTheThread.Execute;
var i: integer;
begin
  i := 0;
  Application.Log(etDebug, 'Thread.Execute');
  try
    repeat
      Sleep(1000); //milliseconds
      inc(i);
      Application.Log(etDebug, 'Thread.Loop ' + Format('Tick :%d', [i]));
    until Terminated;
  finally
    Application.Log(etDebug, 'Thread.LoopStopped');
  end;
end;

destructor TTheThread.Destroy;
```

```
begin
  Application.Log(etDebug, 'Thread.Destroy');
  inherited Destroy;
end;

{$REGION ' - Daemon - '}
function TTheDaemon.Install: boolean;
begin
  result := inherited Install;
  Application.Log(etDebug, 'Daemon.installed: ' + BoolToStr(result));
end;

function TTheDaemon.UnInstall: boolean;
begin
  result := inherited UnInstall;
  Application.Log(etDebug, 'Daemon.Uninstall: ' + BoolToStr(result));
end;

function TTheDaemon.Start: boolean;
begin
  result := inherited Start;
  Application.Log(etDebug, 'Daemon.Start: ' + BoolToStr(result));
  FThread := TTheThread.Create(true);
  FThread.FreeOnTerminate := true;
  FThread.Resume;
end;

function TTheDaemon.Stop: boolean;
begin
  result := inherited Stop;
  Application.Log(etDebug, 'Daemon.Stop: ' + BoolToStr(result));
  FThread.Terminate;
  FThread.WaitFor;
  FThread := nil;
end;

function TTheDaemon.Pause: boolean;
begin
  result := inherited Pause;
  Application.Log(etDebug, 'Daemon.Pause: ' + BoolToStr(result));
  FThread.Suspend;
end;

function TTheDaemon.Continue: boolean;
begin
  result := inherited Continue;
  Application.Log(etDebug, 'Daemon.Continue: ' + BoolToStr(result));
  FThread.Resume;
end;

function TTheDaemon.Execute: boolean;
begin
  result := inherited Execute;
  Application.Log(etDebug, 'Daemon.Execute: ' + BoolToStr(result));
end;

function TTheDaemon.ShutDown: boolean;
begin
  result := inherited ShutDown;
  Application.Log(etDebug, 'Daemon.ShutDown: ' + BoolToStr(result));
end;
{$ENDREGION}

{$REGION ' - DaemonMapper - '}
constructor TTheDaemonMapper.Create(AOwner: TComponent);
begin
  Application.Log(etDebug, 'DaemonMapper.Create');
  inherited Create(AOwner);
  with DaemonDefs.Add as TDaemonDef do
  begin
    DaemonClassName := 'TTheDaemon';
    Name := 'theDaemon';
    Description := 'The Daemon Exsample';
    DisplayName := 'The Daemon';
    RunArguments := '--run';
    Options := [doAllowStop,doAllowPause];
    Enabled := true;
    with WinBindings do
    begin
      StartType := stBoot;
      WaitHint := 0;
```

```
      IDTag := 0;
      ServiceType := stWin32;
      ErrorSeverity := esNormal;//esIgnore;
    end;
//    OnCreateInstance := ?;
    LogStatusReport := false;
  end;
  OnInstall := @Self.ToDoOnInstall;
  OnRun := @Self.ToDoOnRun;
  OnUnInstall := @Self.ToDoOnUninstall;
  OnDestroy := @Self.ToDoOnDestroy;
  Application.Log(etDebug, 'DaemonMapper.Createted');
end;

procedure TTheDaemonMapper.ToDoOnInstall(Sender: TObject);
begin
  Application.Log(etDebug, 'DaemonMapper.Install');
end;

procedure TTheDaemonMapper.ToDoOnRun(Sender: TObject);
begin
  Application.Log(etDebug, 'DaemonMapper.Run');
end;

procedure TTheDaemonMapper.ToDoOnUnInstall(Sender: TObject);
begin
  Application.Log(etDebug, 'DaemonMapper.Uninstall');
end;

procedure TTheDaemonMapper.ToDoOnDestroy(Sender: TObject);
begin
  //doesn't comes here
  Application.Log(etDebug, 'DaemonMapper.Destroy');
end;
{$ENDREGION}

{$R *.res}

begin
  RegisterDaemonClass(TTheDaemon);
  RegisterDaemonMapper(TTheDaemonMapper);
  RegisterDaemonApplicationClass(TCustomDaemonApplication);
  heaptrc.SetHeapTraceOutput(ChangeFileExt(ParamStr(0), '.heap'));
  with Application do
  begin
    Title := 'Daemon Application';
    EventLog.LogType := ltFile;
    EventLog.DefaultEventType := etDebug;
    EventLog.AppendContent := true;
    EventLog.FileName := ChangeFileExt(ParamStr(0), '.log');
    Initialize;
    Run;
  end;
end.
```

Ausgabe:

```
> thedaemon.exe --install
thedaemon [2016-12-01 19:31:02.891 Debug] DaemonMapper.Create
thedaemon [2016-12-01 19:31:02.891 Debug] DaemonMapper.Createted
thedaemon [2016-12-01 19:31:02.891 Debug] DaemonMapper.Install
thedaemon [2016-12-01 19:31:02.907 Debug] Daemon.installed: true

> net start theDaemon
thedaemon [2016-12-01 19:31:05.360 Debug] DaemonMapper.Create
thedaemon [2016-12-01 19:31:05.360 Debug] DaemonMapper.Createted
thedaemon [2016-12-01 19:31:05.360 Debug] DaemonMapper.Run
thedaemon [2016-12-01 19:31:05.360 Info] Daemon TheDaemon current status: Start Pending
thedaemon [2016-12-01 19:31:05.360 Debug] Daemon.Start: true
thedaemon [2016-12-01 19:31:05.360 Info] Daemon TheDaemon current status: Running
thedaemon [2016-12-01 19:31:05.360 Debug] Daemon.Execute: false
thedaemon [2016-12-01 19:31:05.360 Debug] Thread.Execute
thedaemon [2016-12-01 19:31:06.376 Debug] Thread.Loop Tick :1
thedaemon [2016-12-01 19:31:07.379 Debug] Thread.Loop Tick :2
thedaemon [2016-12-01 19:31:08.382 Debug] Thread.Loop Tick :3

> net pause theDaemon
thedaemon [2016-12-01 19:31:08.820 Info] Daemon TheDaemon current status: Pause Pending
thedaemon [2016-12-01 19:31:08.820 Debug] Daemon.Pause: true
thedaemon [2016-12-01 19:31:08.820 Info] Daemon TheDaemon current status: Paused

> net continue theDaemon
```

```
thedaemon [2016-12-01 19:31:12.655 Info] Daemon TheDaemon current status: Continue Pending
thedaemon [2016-12-01 19:31:12.655 Debug] Daemon.Continue: true
theDaemon [2016-12-01 19:31:12.655 Info] Daemon TheDaemon current status: Running
theDaemon [2016-12-01 19:31:13.224 Debug] Thread.Loop Tick :4
theDaemon [2016-12-01 19:31:14.239 Debug] Thread.Loop Tick :5
theDaemon [2016-12-01 19:31:15.255 Debug] Thread.Loop Tick :6
theDaemon [2016-12-01 19:31:16.256 Debug] Thread.Loop Tick :7

> net stop theDaemon
theDaemon [2016-12-01 19:31:16.725 Info] Daemon TheDaemon current status: Stop Pending
theDaemon [2016-12-01 19:31:16.725 Debug] Daemon.Stop: true
thedaemon [2016-12-01 19:31:17.272 Debug] Thread.Loop Tick :8
thedaemon [2016-12-01 19:31:17.272 Debug] Thread.LoopStopped
thedaemon [2016-12-01 19:31:17.272 Debug] Thread.Destroy
theDaemon [2016-12-01 19:31:17.272 Info] Daemon TheDaemon current status: Stopped

> thedaemon.exe --uninstall
theDaemon [2016-12-01 19:31:20.198 Debug] DaemonMapper.Create
theDaemon [2016-12-01 19:31:20.198 Debug] DeamonMapper.Createted
theDaemon [2016-12-01 19:31:20.214 Debug] DeamonMapper.Uninstall
theDaemon [2016-12-01 19:31:20.214 Debug] Daemon.Uninstall: true
```

Autor: Udo Schmal, veröffentlicht: 02.08.2012, letzte Änderung: 03.12.2016