

片桐継(カタギリツグ)はいつも怪しい

- 片桐を知るためのGoogleキーワード
 - SandCastle日本語版
 - マルチスレッドやまたのおろち
 - スレッドプールはキャンセルできない
 - わんくまハムスター
 - 創世のカタギリオン
 - 一万件と二千件あっても動いてる～♪
- その正体は S/人/謎の着物着付け師(見習い)/

Windowsサービス、作ってみたい？

- Agenda

- Windowsサービスって何？
- Windowsサービス作りに大切なこと
- .Netの世界のWindowsサービス
- 作ってみるWindowsサービス
- まとめ

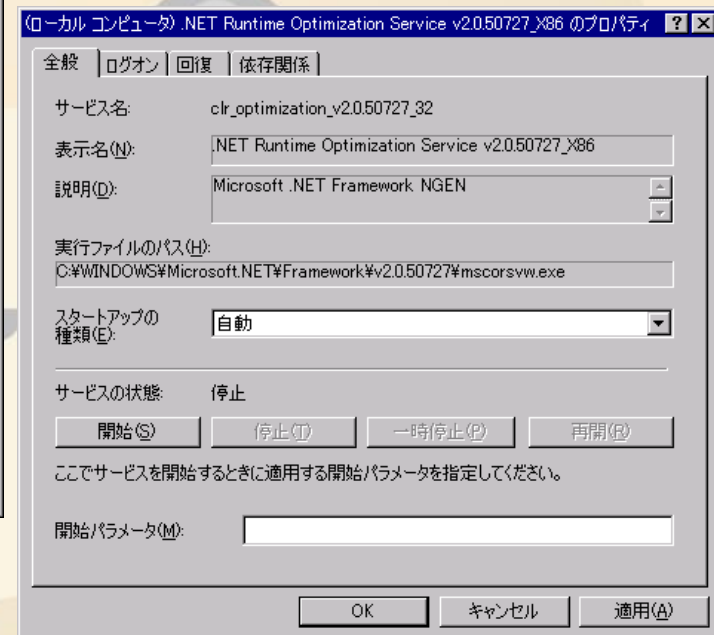
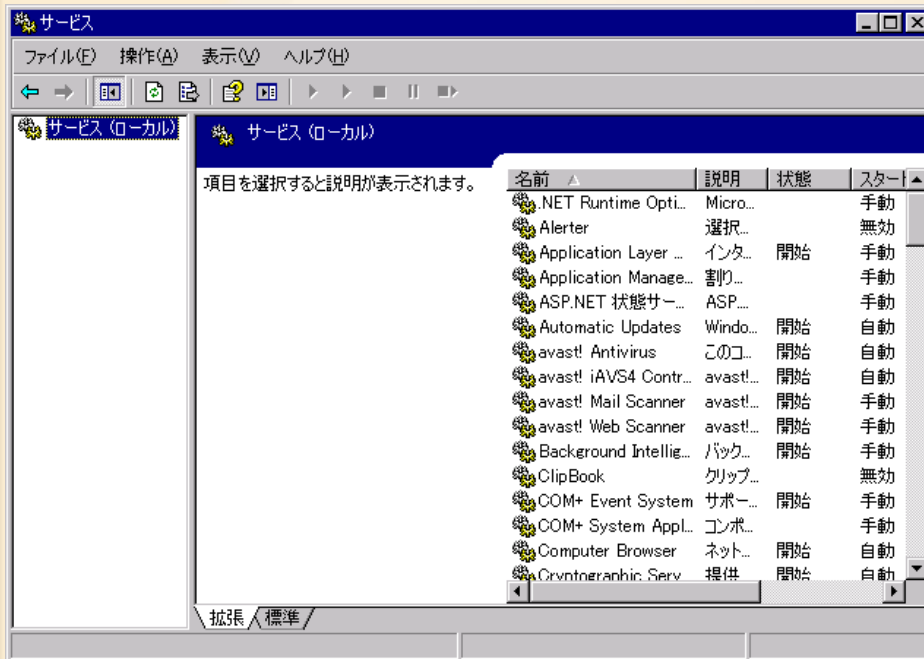
Windowsサービスって何？

Microsoft Windows オペレーティングシステム (OS) で長時間動作し、ユーザーとのやりとり無しで特定機能を実行するものである。WindowsサービスはOSの起動時に起動するよう設定でき、Windows が動作中はずっとバックグラウンドで動作する。あるいは、手動で要求したときに動作するようにも設定できる (By Wikipedia)

- 要するに何もしなくてもスイッチいれてOSが立ち上がったから勝手に動いてるプログラム
- PC触っている分には動いているのが見えないので、何やってんのかさっぱり判らないプログラム

どこで判るの？

- 「管理ツール」の「サービス」をクリックすることでできます



どうやって動かすの？

- サービスの開始
 - サービス画面で開始ボタン
- サービスの停止
 - サービス画面で停止ボタン
- 他にも、net コマンド(他のPCのサービス用)や sc コマンド(自分のPCのサービス用)なんかでもできる

その正体は？

- Exe形式のプログラムアプリケーション
 - でも、どんなexeでもいいのかというとそうじゃない
 - コンソール画面が使えない
 - Formが使えない
 - プロセス停止すると復旧させる手段がない
 - PC再起動が一般的

Windowsサービスの動く仕組みを見てみよう

Windowsサービスの動く仕組み

普段、私たちはこっちを操作



サービス
コントロール

こっちなでなにやってるかはサッパリ判らない
窓口経由でないと、何も指示できない



アプリケーション

アプリケーション

サービスアプリ

OS

この間には越えられない絶対的な壁がある

Windowsサービスサービス作りに大切なこと

- 知っておきたいキーワード
 - イベントログ(イベントビューア)
 - アプリケーション応答制限時間

イベントログ(イベントビューア)

- サービスの動作のログを書き込みます
 - サービスの開始と停止の情報を自動的に書かせることができます
 - どの権限で動作しても、サービスプログラムから自由に書き込む事ができる場所です

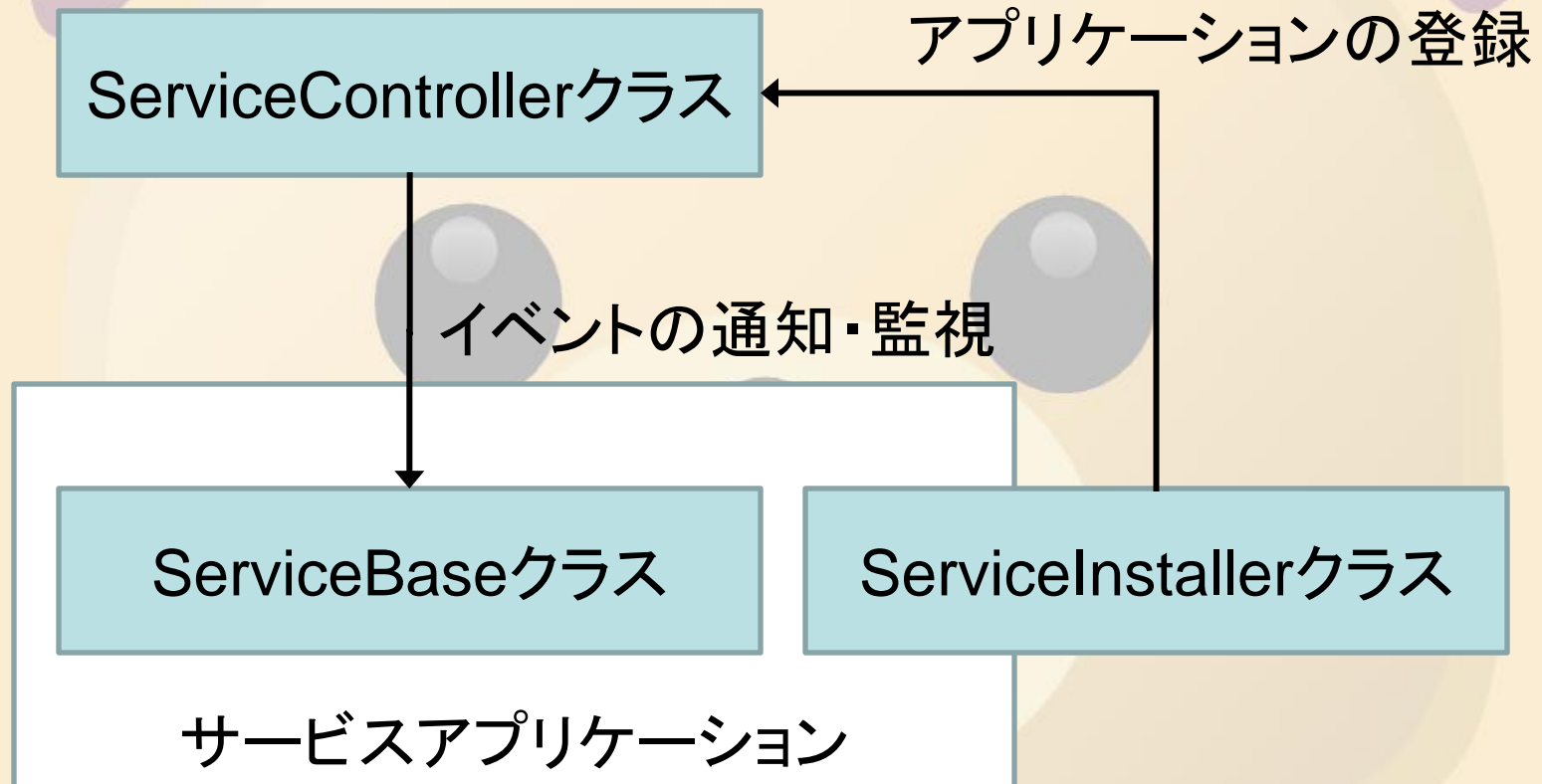


アプリケーション応答制限時間

- サービスコントローラからイベント通知されると、30秒以内にプロセスが応答(イベント処理終了通知)をしなければなりません。
 - 30秒以内に通知に対する応答がないと、サービスは例外と判断、プロセスを強制停止します。
 - つまり、窓口で顧客から何かを依頼されたら、中の人には30秒以内に窓口に戻答をしなくてはなりません。
 - 中の人から窓口にお願ひされてから、他の人に「これお願ひね」と頼んで、窓口で「お願ひしといたよー」と返事するだけでもOK

.NetFrameworkの世界のWindowsサービス

- クラス関係図



クラス間の微妙な力関係

- ServiceControllerからServiceBaseへイベント通知できる
- ServiceControllerはServiceBaseの動きを監視できる
- ServiceBaseからServiceControllerへの通知・監視・制御はできない(推奨されていない)
- ServiceInstallerは、サービスをインストールする時にだけ必要なクラス(実行には不要)

.Netで作ったServiceが動くまでに必要な手続き

- ビルドしたExeをServiceInstallerがサービスに登録
- ServiceControllerクラスからそのアプリケーションを起動
 - 以下、ServiceBaseクラスのお仕事
 - プロセス開始時にInitializeメソッドが動き、インスタンスを作成
 - 通知イベントに応答するためにMTA(マルチスレッドアパートメント)でプロセスをスタート
 - 開始通知イベントに応答→応答処理完了
 - ServiceBaseクラスのお仕事は30秒以内
- ServiceControllerがサービスのスタートを確認
 - 30秒以内に応答なければ失敗と判断

作ってみるWindowsサービス

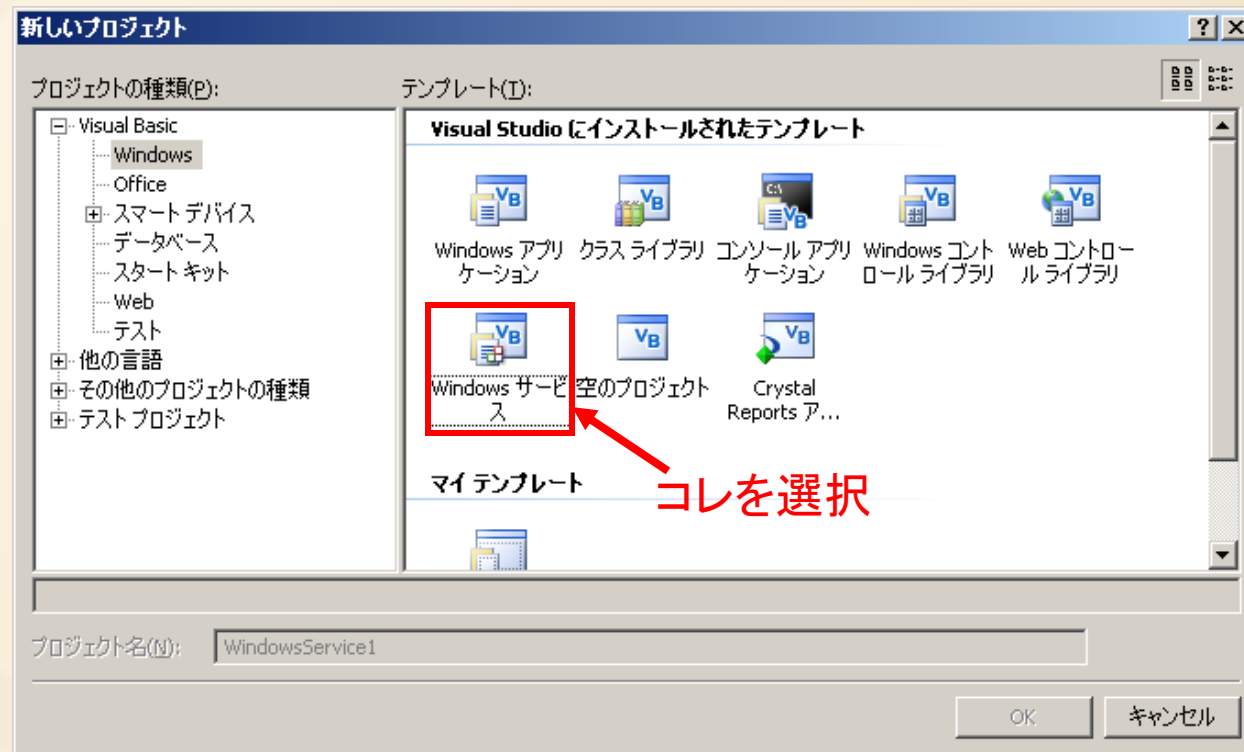
- ServiceBaseクラスを継承して作ります。
 - サービスアプリケーション自身を作る場合にはServiceControllerは使いません。

作るときのキーワード

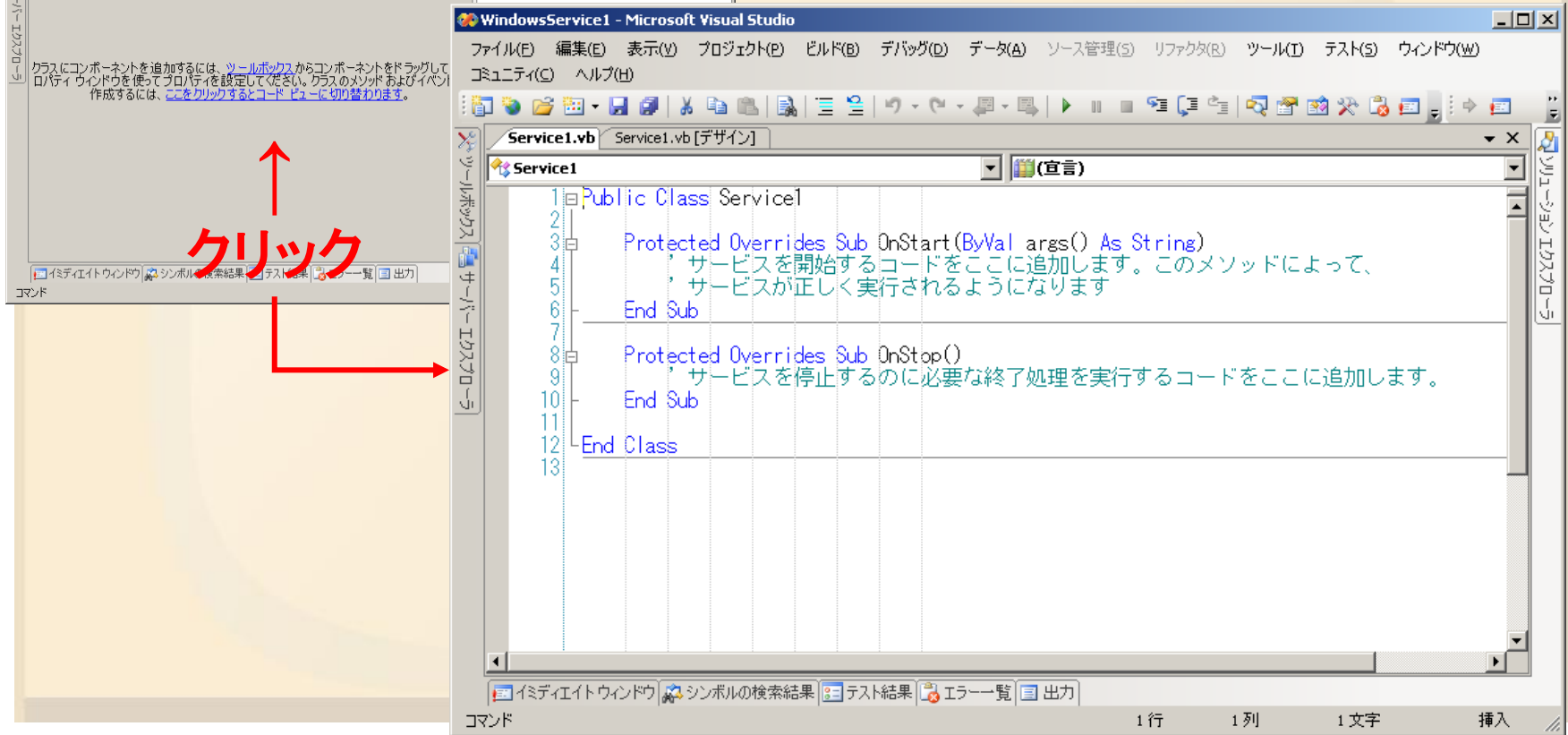
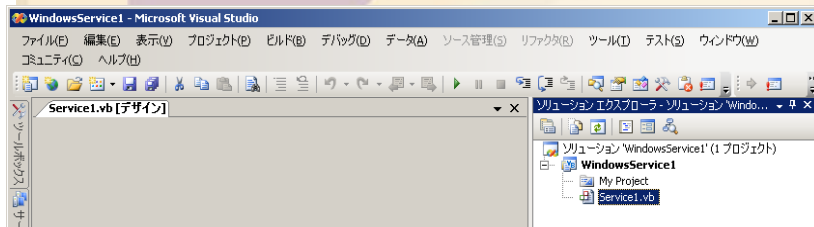
- **まずはこの二つのメソッドを覚えよう**
 - OnStart
 - 開始ボタンで通知されてくるイベント
 - できること
 - » モニター用スレッドの起動／待機用スレッドの起動
 - してはいけないこと
 - » サービスプロセスの停止／プロセスの強制終了／時間のかかるループ処理、同期処理
 - OnStop
 - 停止ボタンで通知されてくるイベント
 - できること
 - » 子スレッドの停止／サービスプロセスの停止／プロセスの強制終了
 - してはいけないこと
 - » サービスプロセスの継続

これらのメソッドは必ず30秒以内に終了できるように(ry

まずは新規プロジェクトを作成



こういうのが出来上がる



こんなカンジを作ってみた

```
WindowsService1 - Microsoft Visual Studio
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) データ(A) ソース管理(S) リファクタ(R) ツール(T) テスト(S) ウインドウ(W) コミュニティ(C) ヘルプ(H)
Service1.vb Service1 [デザイン]
Service1
Public Class Service1
    Private th As New Threading.Thread(AddressOf Me.MoniterThread)

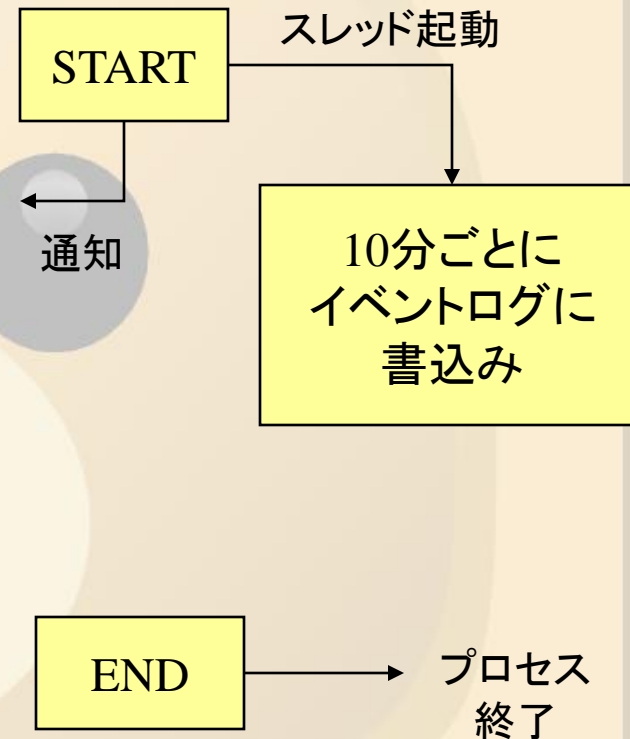
    Protected Overrides Sub OnStart(ByVal args() As String)
        ' サービスを開始するコードをここに追加します。このメソッドによって、
        ' サービスが正しく実行されるようになります
        Me.EventLog.WriteEntry("ほな、はじめるでえ")
        th.Start()
    End Sub

    Protected Overrides Sub OnStop()
        ' サービスを停止するのに必要な終了処理を実行するコードをここに追加します。
        Me.EventLog.WriteEntry("ほな、さいなら〜")
        Try
            th.Abort()
        Catch ex As Threading.ThreadAbortException
            Exit Try
        End Try
        Me.Stop()
    End Sub

    Private Sub MoniterThread()
        Dim sleepTime As New TimeSpan(0, 1, 0)

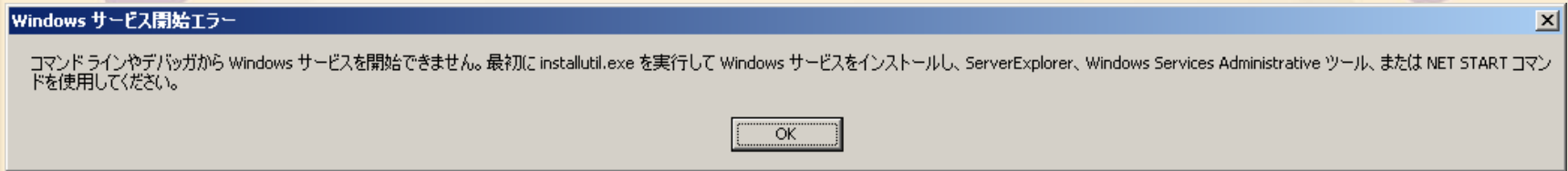
        While (True)
            If (Now.Minute Mod 10) = 0 Then
                Me.EventLog.WriteEntry(String.Format("[0]時やでえ", Now.ToLongTimeString))
            End If

            Threading.Thread.Sleep(sleepTime)
        End While
    End Sub
End Class
```



どうやって動かすの？

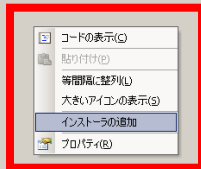
- いつものように動かそうと「▼実行」すると



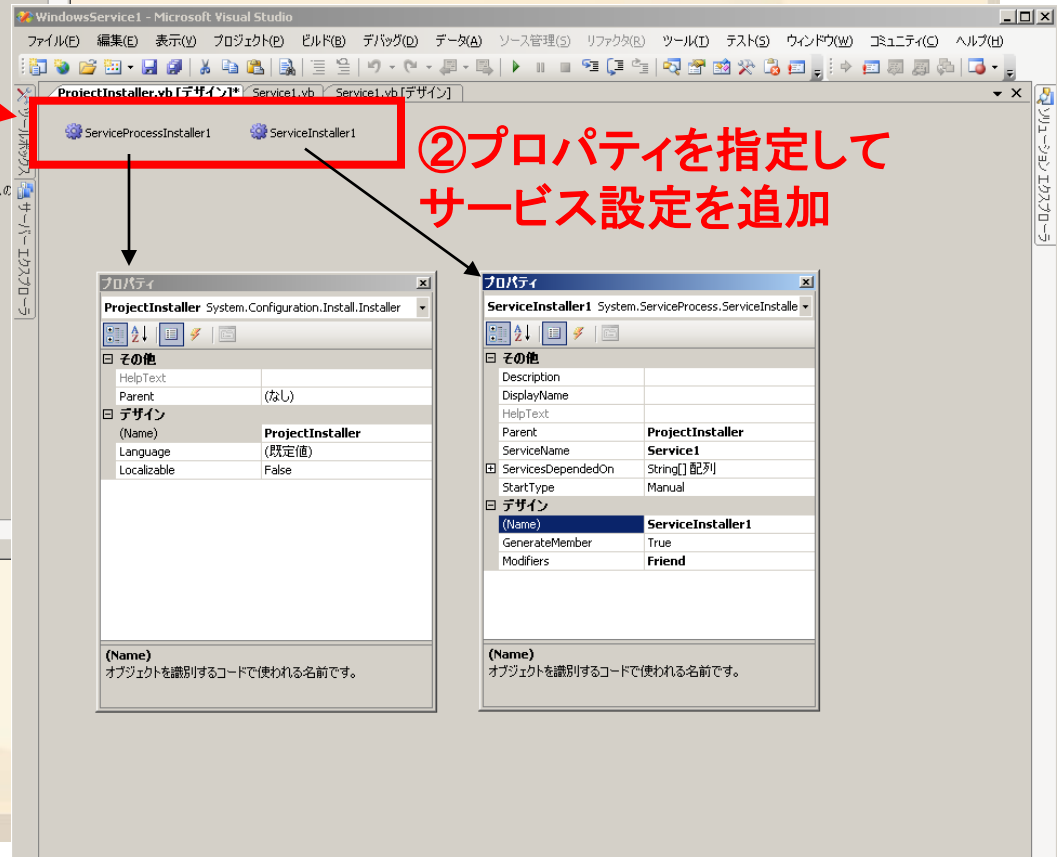
- サービスプログラムは、インストーラーを使ってサービスに登録してからでないと Visual Studio 上でデバッグできません

サービスインストーラを設定する

①右クリック



②プロパティを指定して
サービス設定を追加



③設定できたらビルド



インストールしてみよう

- MSIを作ってインストール
 - セットアッププロジェクトを作ってインストール、楽
 - インストール情報を変えたいときに面倒
- InstallUtil.exeを使ってインストール
 - 実行パス指定が面倒
 - 置換えが楽
- SCバッチコマンドでインストール
 - コマンド文字列が長くて作るのが面倒
 - 置換えが楽

どうやってデバッグするの？

- VisualStudioからアタッチしよう

The screenshot shows the Visual Studio IDE with the 'Attach to Process' dialog box open. The dialog box has a 'Transport' dropdown set to 'Default', a 'Filter' dropdown, and a 'Attach first' dropdown set to 'Automatic: Manage Code'. The 'Attach to Process' button is highlighted with a red box. The 'Attach to Process' dialog box also has a 'Refresh' button and 'Attach (A)' and 'Cancel' buttons at the bottom.

プロセスにアタッチ

トランスポート(T): 既定値

修飾子(O): 参照(B)...

トランスポート情報
既定のトランスポートでは、このコンピュータまたは Microsoft Visual Studio リモート デバッグ モニタ (MSYSOMN.EXE) を実行しているリモートコンピュータ上のプロセスを選択できます。

アタッチ先: 自動: マネージコード 選択(S)...

選択可能なプロセス(O)

プロセス	ID	タイトル	型	ユーザー名	セッション
MAPISP32.EXE	3168		x86		0
Mctray.exe	2692		x86		0
OUTLOOK.EXE	2752		x86		0
POWERPNT.EXE	3812		x86		0
SCAuto.exe	2256		x86		0
shstat.exe	4480		x86		0
TSVNCache.exe	2424		x86		0
UdaterUI.exe	2472		x86		0
VISIO.EXE	164		x86		0
WindowsService1....	3744	Windows サービス開始エラー	マネージ		0
WinShot.exe	3156		x86		0

コレを選んで「アタッチ」ボタン

更新(B)

すべてのユーザーからのプロセスを表示する(L) すべてのセッションのプロセスを表示する(N)

アタッチ(A) キャンセル



まとめ

- .NetFrameworkによって、サービスアプリケーション開発はぐっと楽になった
- けれど開発する上で制限、キーワード、クラス関係の理解は必須
- 使い方や用途は色々とあります。ぜひ、挑戦してみてくださいね！

参考資料

- Windows サービス アプリケーション (MSDN)
 - [http://msdn.microsoft.com/ja-jp/library/y817hyb6\(VS.80\).aspx](http://msdn.microsoft.com/ja-jp/library/y817hyb6(VS.80).aspx)
- Visual C# in a Windows Service application for a setup project create to how
 - <http://support.microsoft.com/kb/816169/ja>
- VB.net Windows サービス 開発 テクニック (CQ 出版)
 - ISBN4-7898-1837-3
- Programming Windows Services with Microsoft Visual Basic 2008 (microsoft press)
 - ISBN13-978-0-7356-2433-7
 - ISBN10-0-7356-2433-X