

NEWTONE

プロテクトコンポーネント

ProtectKit3.5

プログラミングマニュアル

目次

1. 動作環境.....	3
2. お問い合わせについて.....	4
3. インストール.....	5
4. サンプルプログラムについて.....	7
5. プロテクトアプリケーションの運用手順.....	8
5-1. パスワード発行型プロテクトの場合(ネイティブアプリケーション).....	8
5-2. パスワード発行型プロテクトの場合(CLR アプリケーション).....	10
5-3. ネットワーク型プロテクトの場合(ネイティブアプリケーション).....	14
6. プロテクト情報設定ツールについて.....	19
6-1. パスワード発行型プロテクトの場合(ネイティブアプリケーション).....	19
6-2. パスワード発行型プロテクトの場合(CLR アプリケーション).....	24
6-3. ネットワーク型プロテクトの場合(ネイティブアプリケーション).....	29
7. パスワード発行について.....	31
7-1. 付属の実行型プログラムを使用する方法(パスワード発行ツール).....	32
7-2. 付属の DLL を使用する方法(パスワード発行 DLL).....	33
7-3. 付属の OCX を使用する方法(パスワード発行 OCX).....	36
8. 運用上の注意事項.....	45
9. 作成したアプリケーションの配布.....	47
10. DdScm.exe,DdScm32.exe,DdScm64.exe のオプション機能について.....	49
11. ProtectKit2/ProtectKit2.5 との互換性.....	50
12. CLR アプリケーションのビルドオプション.....	51
索引.....	52

※表記中の社名、製品名などは各社の商標または登録商標です。

※本仕様、及び価格などは予告なしに変更する場合があります。

1. 動作環境

1) 対応プラットフォーム

ProtectKit.NET:

Windows 2000/XP/Vista/7, Windows Server 2003/2008
.NET Framework 2.0 以上

ProtectKit3:

Windows 2000/XP/Vista/7, Windows Server 2003/2008

2) 対応開発コンテナ

ProtectKit.NET:

Visual Studio 2005/2008/2010

*Visual C++のアンマネージドコードには対応していません。

ProtectKit3:

Visual Basic 6.0、Visual C++ 2010/2008/2005/2003/6.0、Embarcadero RAD Studio 2010(C++Builder 2010/Delphi 2010)、
CodeGear RAD Studio 2009(C++Builder 2009/Delphi 2009 for Win32)、CodeGear RAD Studio 2007(C++Builder 2007/Delphi
2007 for Win32)、Borland Developer Studio 2006(C++Builder/Delphi for Win32)、Delphi 2005(Win32)/7/6/5、C++Builder
6/5

*Visual Studio .NET のマネージドコードには対応していません。

※すべて日本語環境です。

2. お問い合わせについて

弊社にお問い合わせされる場合は、以下の項目をお知らせいただきますようお願いいたします。

ユーザ情報

- 1) ユーザ ID (UID-XXXXXX)
- 2) お名前 (法人登録の場合は、会社名も併せてお知らせください。)
- 3) ユーザ登録している電話番号、FAX 番号もしくはメールアドレス
- 4) 有償サポート契約を結ばれている方は契約番号

製品に関する情報

- 1) 製品名
- 2) バージョン
- 3) 製品のシリアル No.
- 4) ocx または dll ファイルのタイムスタンプ

開発環境に関する情報

- 1) OS
- 2) 開発コンテナ
- 3) 利用形態 (.NET Framework コンポーネント or ActiveX or DLL)

ご質問の内容

どのようにしたらその症状が発生するのか、あるいはどのような事柄が聞きたいのか、できるだけ簡潔にご説明ください。口頭や文章で説明するのが難しい場合は、再現性のあるプログラムをメールにて添付していただいた方がよりスムーズに解決する場合もございますので、その場合はプログラムをLZHもしくはZIP形式で圧縮していただき、お送りください。

また、製品とは関係のないプログラミング技法や WindowsAPI などに関するご質問にはお答えいたしかねますので、予めご了承ください。

問い合わせ先

「お使いになる前に」(パッケージ版は同梱の折込案内、ダウンロード版は別の PDF ファイル)のサポートについてをご覧ください。

注意

1PC 開発ライセンスはユーザ登録された方がサポート対象となります。

開発ライセンスパックはユーザ登録された方もしくは同一法人の方(法人登録の場合)がサポート対象となります。

3. インストール

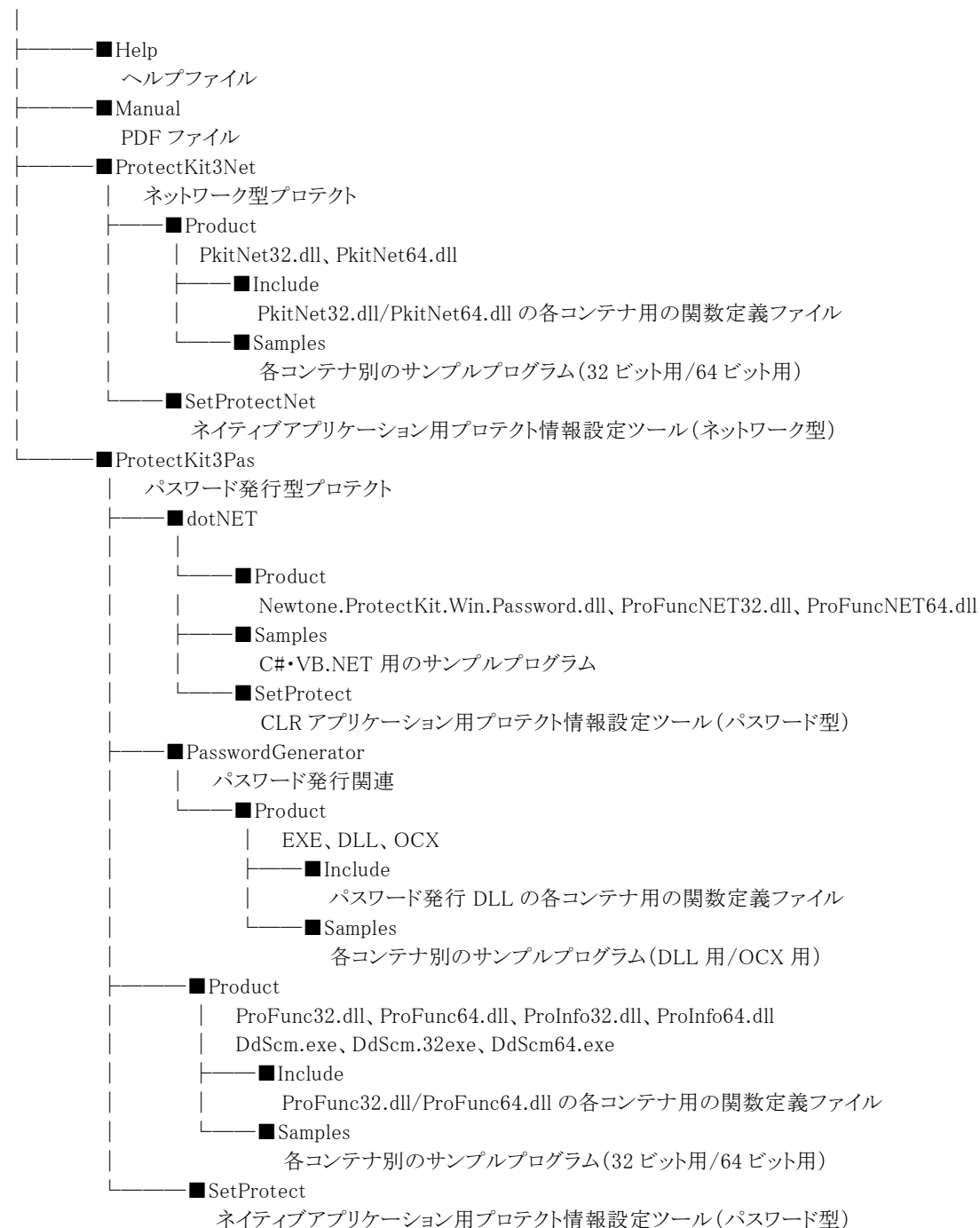
●インストール方法

CD-ROM 内の「ProtectKit35.exe」を実行して、その後は指示されるメッセージに従ってインストールを進めてください。

●インストール後のフォルダの構成

ProtectKit3.5 のインストール後のフォルダ構成は以下の通りです。

■インストール時指定したフォルダ



●パスワード発行 OCX を使用する場合

パスワード発行 OCX を使用する場合は、OCX をレジストリに登録してから開発環境への組み込み(ActiveX コントロールの組み込み/登録)が必要になります。

以下に VisualBasic6.0 と Delphi5 について説明いたします。

インストール

(1)VisualBasic 6.0 の場合

- 1.「プロジェクト(P)」メニューから「コンポーネント(O)」を選択します。
- 2.「コントロール」タブから次の項目を選択して「OK」ボタンを押します。
[Pr3PassGen Control]
- 3.ツールボックスに組み込んだコントロールのアイコンが表示されます。

(2)Delphi5 の場合

- 1.「コンポーネント(C)」メニューから「ActiveX コントロールの取り込み(X)」を選択します。
- 2.「ActiveX の取り込み」から次の項目を選択して「インストール」ボタンを押します。
[Pr3PassGen Control]
- 3.コンポーネントパレットに組み込んだコントロールのアイコンが表示されます。

詳しくは各開発コンテナのマニュアルなどをご覧ください。

●レジストリの登録及び解除について

手で OCX をレジストリに登録したり解除したりする場合は、

(登録)「REGSVR32 "`<path>¥Pr3PassGen.ocx`"

(解除)「REGSVR32 /U "`<path>¥Pr3PassGen.ocx`"

と実行します。

※`<path>`は OCX が存在するパスを指します。

●.NET Framework コンポーネントを使用する場合

Visual Basic, Visual C#, Visual C++の場合

- 1.「ツール(T)」メニューから「ツールボックス アイテムの選択(X)」を選択します。
- 2.「.NET Framework コンポーネント」タブを選択し、参照ボタンを押します。
"Newtone.ProtectKit.Win.Password.dll"を選択してから「開く」ボタンを押します。
"Newtone.ProtectKit.Win.Password.dll"が追加されたことを確認してから「OK」ボタンを押します。
- 3.ツールボックスに"PasswordProtect"のアイコンが表示されます。

4. サンプルプログラムについて

インストール時指定したフォルダの階層化の¥Samples の中に、対応コンテナ別のフォルダがあります。
その中のプロジェクトファイルを使用して、動作を確認してください。
それぞれのサンプルの簡単な説明としてテキストファイル(txt)をご用意しましたので、実行前にご覧ください。
また、サンプルプログラムとして、パスワード発行型プロテクト・ネットワーク型プロテクトのサンプル及びパスワード発行の DLL やパスワード発行 OCX を使用したサンプルをご用意しました。

5. プロテクトアプリケーションの運用手順

ProtectKit3.5の運用方法として、パスワードを発行するパスワード発行型プロテクトとネットワーク(LAN)上でアプリケーションの使用状況を検出するネットワーク型プロテクトがあります。それぞれの方法について、使い方の簡単な流れを説明します。

5-1. パスワード発行型プロテクトの場合(ネイティブアプリケーション)

STEP-1 プログラムの作成

[プロテクト部分のサンプルコード例(VisualBasic)]

■初期化処理

```
Private Sub Form_Load()
    Dim Lockword As String * 9
    Dim ChkLockword As String
    Dim LockwordLen As Long
    ...
    ret = PK3Initialize("Project1.exe", "ProInfo32.dll")
    If ret <> 0 Then
        If ret = 1 Then Msg= "ProtectKit が作動していません"
        If ret = 2 Then Msg= "プロテクトファイル読込エラー"
        If ret = 3 Then Msg= "ロックワードが不正です"
        ...
    End
End If

LockwordLen = PK3GetLockword(Lockword)
ChkLockword = "A" & "b" & "c" & "d" & "e" & "f" & "g" & "H"
If LockwordLen <> 8 Or ChkLockword <> Left(Lockword, LockwordLen) Then
    MsgBox "ロックワードが不正です。"
End
End If
Label1.Caption = "OK"
End Sub
```

■アプリケーションが動作するPC情報の取得

```
Private Sub Command1_Click()
    Dim str As String * 64
    Dim strLen As Long
    ret = PK3QueryPcID(str, strLen)
    Text2.Text = str
    ...
End Sub
```

■パスワードの設定

```
Private Sub Command2_Click()
    ret = PK3SetPassword(Text1.Text)
End Sub
```

■プロテクトのチェック

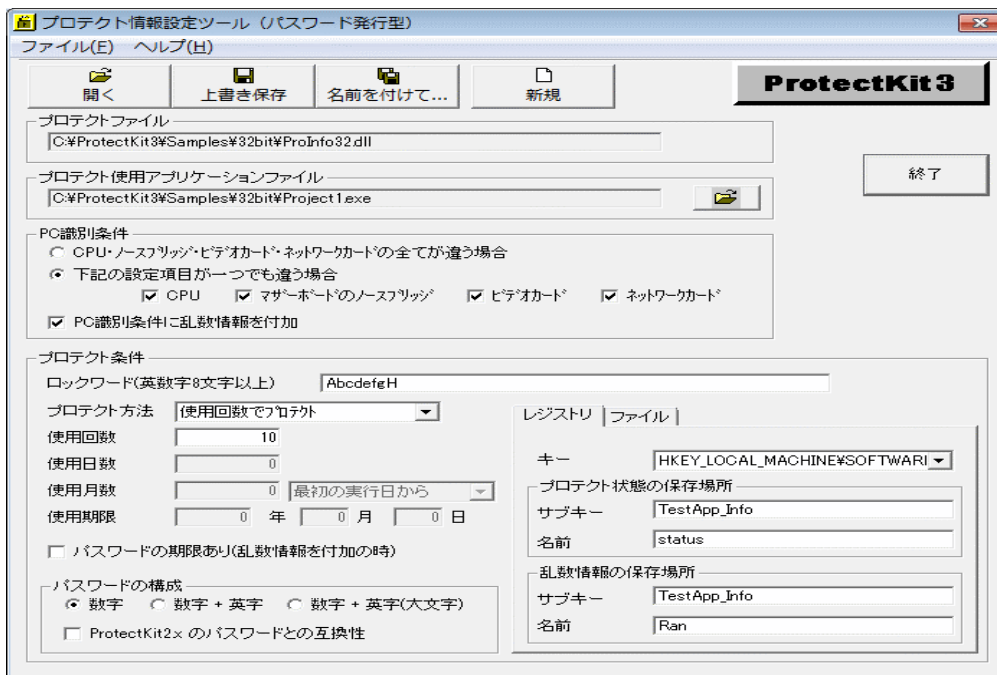
```
Private Sub Command3_Click()
    ret = PK3QueryProtect()
    If ret = 0 Then Msg = "プロテクトなし"
    If ret = 1 Then Msg = "プロテクト中"
    If ret = 2 Then Msg = "使用回数オーバー"
    ...
End Sub
```

※詳しくは付属のサンプルプログラムをご覧ください。

STEP-2 プロテクト条件の設定

[プロテクト設定情報ツール(パスワード発行型)の利用]

作成したアプリケーションファイルとプロテクト情報保持ファイルにプロテクト情報を設定します。(アプリケーションを再構築した場合は再度プロテクト情報を設定する必要があります。)



※パスワードは外部に流出しないようにお取り扱いに十分ご注意ください。

■プロテクト方法の種類

パスワード発行型でのプロテクトの方法は次の通りです。
(「プロテクト方法」は「プロテクト条件」に含まれます。)

プロテクト方法	アプリケーションの動作
1.標準プロテクト	パスワード発行を受けない限り使用できない
2.使用回数プロテクト	設定回数分の起動が可能
3.使用日数プロテクト	設定日数分の使用が可能
4.使用月数プロテクト	設定月数分の使用が可能
5.使用期限プロテクト	設定期限までの使用が可能
6.パスワード有効期限 設定時	上記 2～4 と併用することで有効期限後は別のパスワードが必要になる

STEP-3 デジタル署名の添付

作成したアプリケーションファイルにデジタル署名を添付します。デジタル署名が不要な場合は STEP-4 へ。

STEP-4 アプリケーションの配布

開発環境付属(または市販)のインストーラを利用して、完成したアプリケーションと関連ファイルを配布します。
※配布ファイルについては、「9. 作成したアプリケーションの配布」をご覧ください。

STEP-5 パスワードの取得と通知

[パスワード発行機能の利用]

■パスワード発行ツールの利用

STEP-2 で設定したパスワードとエンドユーザから通知された PCID を使い、エンドユーザに発行するパスワードを取得します。取得したパスワードをエンドユーザへ通知します。



■パスワード発行 OCX、DLL の利用

「パスワード発行ツール」と同等な機能を有する OCX、DLL を提供。これらを API 関数として利用することでパスワードをプログラム上で取得でき、ユーザ管理などの社内システムとの連動も可能になります。また、OCX は IIS と ASP を使った Web アプリケーションからの呼び出しもサポートしています。さらに「パスワード発行ツール」には無い E-Mail 送信機能も API 関数として用意してあります。たとえばアプリケーションベンダがエンドユーザへのパスワードを E-Mail で自動送信するシステムを作成する場合に利用できます。

●ASP コード例 (VBScript)

```
Set objCOM = Server.CreateObject("Pr3PassGen.PassGen")
objCOM.szLockword = varLockword
objCOM.szPCID = varPCID
objCOM.lPasswordType = varType
varUnlockword = objCOM.PS3GetPassword
Set objCOM = nothing
```

●E-Mail 送信機能 API 関数利用コード例 (VisualBasic)

```
ret = PK3SendMail("msv.newtone.co.jp", "info@newtone.co.jp", _
    "NEWTONE", "paul@company.com", "Users", _
    "PassWord Information", "パスワードをご連絡いたします…")
```

※パスワード通知については、「7. パスワード発行について」をご覧ください。

5-2. パスワード発行型プロテクトの場合 (CLR アプリケーション)

STEP-1 プログラムの作成

[プロテクト部分のサンプルコード例 (Visual Basic)]

■初期化処理

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    passwordProtect1.DllFileName = "ProFuncNET32.dll"
    Dim ret As Integer = passwordProtect1.Initialize("VB_Sample1.exe", "ProInfo32.dll")
    If (ret <> 0) Then
        Select Case (ret)
            Case -1 : MessageBox.Show("使用できません。")
            Case 1 : MessageBox.Show("サービスが動作していません。")
            Case 2 : MessageBox.Show("情報保持ファイルの読み込みエラーです。")
            Case 3 : MessageBox.Show("ロックワードが不正です。")
            Case 4 : MessageBox.Show("ローカルディスクで実行してください。")
            Case 5 : MessageBox.Show("エラーです。")
        End Select
        Me.Close()
    End If

    Dim s As String = passwordProtect1.GetLockword()
    Dim ChkLockword() As String = "AbcdefgH"
    If (s <> ChkLockword) Then
        MessageBox.Show("ロックワードが不正です。")
        Me.Close()
    End If
    label1.Text = "OK"
```

```

End Sub
■アプリケーションが動作する PC 情報の取得
Private Sub button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
button1.Click
    textBox1.Text = passwordProtect1.QueryPcID()
End Sub
■パスワードの設定
Private Sub button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
button2.Click
    passwordProtect1.SetPassword(textBox2.Text)
End Sub
■プロテクトのチェック
Private Sub button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
button3.Click
    Dim ret As Integer = passwordProtect1.QueryProtect()
    Select Case ret
        Case -1 : label4.Text = "使用不可"
        Case 0 : label4.Text = "プロテクトなし"
        Case 1 : label4.Text = "プロテクト中"
        Case 2 : label4.Text = "使用回数オーバー"
        Case 3 : label4.Text = "使用日数オーバー"
        Case 4 : label4.Text = "使用月数オーバー"
        Case 5 : label4.Text = "使用期限オーバー"
        Case 6 : label4.Text = "日付を戻した"
        Case 7 : label4.Text = "PC が違う"
        Case 8 : label4.Text = "プロテクト情報保持ファイルが正しくない"
        Case 9 : label4.Text = "ロックワードが正しくない"
        Case 10 : label4.Text = "パスワードが正しくない"
        Case 11 : label4.Text = "その他"
        Case 12 : label4.Text = "不正に未使用のプロテクト情報保持ファイルを使用した"
    End Select
    '他の処理
End Sub
※詳しくは付属のサンプルプログラムをご覧ください。

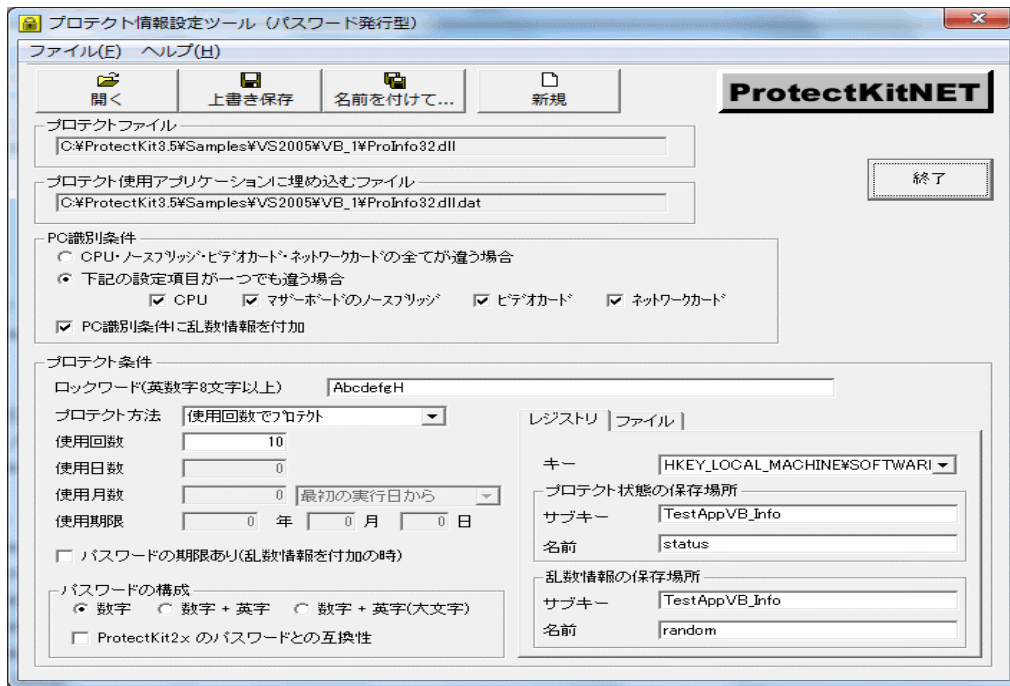
```

STEP-2 プロテクト条件の設定

[プロテクト設定情報ツール(パスワード発行型)の利用]

プロテクト情報保持ファイルとアプリケーションに埋め込むファイルにプロテクト情報を設定します。

プロテクトアプリケーションの運用手順



※パスワードは外部に流出しないようにお取り扱いに十分ご注意ください。

■プロテクト方法の種類

パスワード発行型でのプロテクトの方法は次の通りです。
(「プロテクト方法」は「プロテクト条件」に含まれます。)

プロテクト方法	アプリケーションの動作
1.標準プロテクト	パスワード発行を受けない限り使用できない
2.使用回数プロテクト	設定回数分の起動が可能
3.使用日数プロテクト	設定日数分の使用が可能
4.使用月数プロテクト	設定月数分の使用が可能
5.使用期限プロテクト	設定期限までの使用が可能
6.パスワード有効期限 設定時	上記 2～4 と併用することで有効期限後は別のパスワードが必要になる

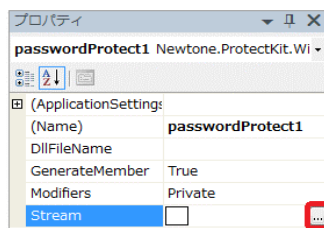
STEP-3 アプリケーションに埋め込むファイルをアプリケーションに設定

STEP-2 で作成したアプリケーションに埋め込むファイルを PasswordProtect クラスの Stream プロパティに設定します。

設定方法:

ここでは 2 種類の方法について説明します。

1.アプリケーションのプロジェクトを開き、PasswordProtect クラスを選択し、プロパティウィンドウの Stream の右のボタン(下の図の赤枠)を押して埋め込むファイルを開きます。



2.アプリケーションのプロジェクトを開き、埋め込むファイルをプロジェクトに追加し、ビルドアクションを「埋め込まれたリソース」とします。

2 の場合、埋め込むファイルをそのまま追加するよりは、別のファイルの一部として取り込み、実行時にその情報を取得する方がよいでしょう。取得した情報は Stream プロパティに設定してください。

STEP-4 デジタル署名の添付

作成したアプリケーションファイルに難読化を施します。難読化が不要な場合は STEP-5 へ。

難読化を行わない場合、プロテクトのチェック部分などのプログラムコードが解析ツールにより明らかになる可能性がありますので、アプリケーションの難読化を行うことをお勧めいたします。難読化を行う際には文字列およびリソースについても暗号化処理を行ってください。

STEP-5 デジタル署名の添付

作成したアプリケーションファイルにデジタル署名を添付します。デジタル署名が不要な場合は STEP-6 へ。

STEP-6 アプリケーションの配布

開発環境付属(または市販)のインストーラを利用して、完成したアプリケーションと関連ファイルを配布します。

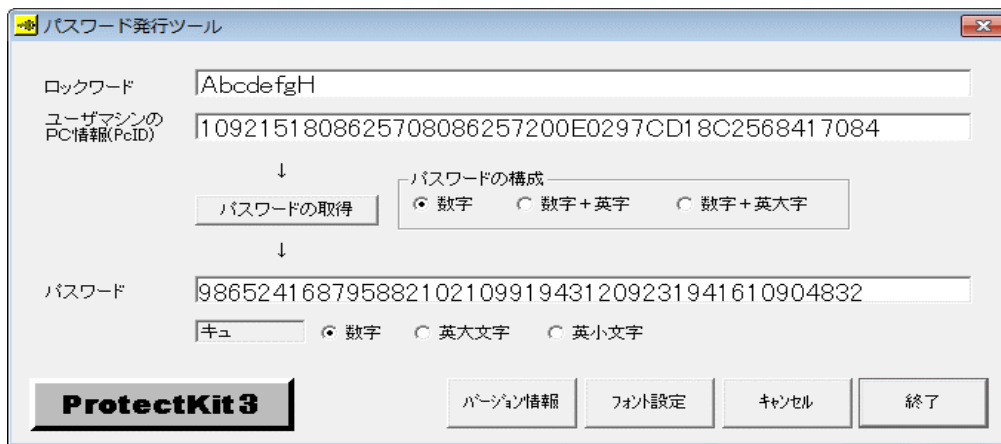
※配布ファイルについては、「9. 作成したアプリケーションの配布」をご覧ください。

STEP-7 パスワードの取得と通知

[パスワード発行機能の利用]

■パスワード発行ツールの利用

STEP-2 で設定したロックワードとエンドユーザから通知された PCID を使い、エンドユーザに発行するパスワードを取得します。取得したパスワードをエンドユーザへ通知します。



■パスワード発行 OCX、DLL の利用

「パスワード発行ツール」と同等な機能を有する OCX、DLL を提供。これらを API 関数として利用することでパスワードをプログラム上で取得でき、ユーザ管理などの社内システムとの連動も可能になります。また、OCX は IIS と ASP を使った Web アプリケーションからの呼び出しもサポートしています。さらに「パスワード発行ツール」には無い E-Mail 送信機能も API 関数として用意してあります。たとえばアプリケーションベンダがエンドユーザへのパスワードを E-Mail で自動送信するシステムを作成する場合に利用できます。

●ASP コード例 (VBScript)

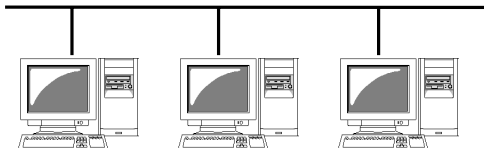
```
Set objCOM = Server.CreateObject("Pr3PassGen.PassGen")
objCOM.szLockword = varLockword
objCOM.szPCID = varPCID
objCOM.lPasswordType = varType
varUnlockword = objCOM.PS3GetPassword
Set objCOM = nothing
```

●E-Mail 送信機能 API 関数利用コード例 (VisualBasic)

```
ret = PK3SendMail("msv.newtone.co.jp", "info@newtone.co.jp", _
    "NEWTONE", "paul@company.com", "Users", _
    "PassWord Information", "パスワードをご連絡いたします…")
```

※パスワード通知については、「9. パスワード発行について」をご覧ください。

5-3. ネットワーク型プロテクトの場合(ネイティブアプリケーション)



ProtectKit3 は、ネットワーク(LAN)上で動作している同一アプリケーションの使用状況を検出します。アプリケーションベンダは不正使用を検出した場合のふるまいをコードで予め記述することができます。たとえば検出後、その不正使用した PC からは、たとえネットワークカードをはずしても、そのアプリケーションを使用できないように設定することもできます。(ProtectKit3 のレジストリ関数を使用します。)また不正使用の検出には、アプリケーションのシリアル番号などによるチェックの他に、アプリケーションが動作している PC の台数を取得することもできます。そうすることで、同時使用ライセンス数による配布契約形態を実現することもできます。ネットワークプロトコルは TCP/IP に対応しています。

STEP-1 プログラムの作成

[プロテクト部分のサンプルコード例(VisualBasic)]

■初期化処理

```
Private Sub Form_Load()
    Dim Network As String * 65
    Dim NetworkLen As Long
    Dim net As PROTECTKIT3NET
    Dim ret As Long
    net.szAppID = "12345678.VB6" & Chr$(0) ' アプリケーション固有のIDを設定
    net.szMessage = Text1.Text & Chr$(0) ' 送信メッセージ(製品のシリアル番号など)
    net.lSendTime = Val(Text3.Text) ' インターバルタイム(秒単位)
    net.hWndHandle = Me.hWnd ' ウィンドウハンドル
    ret = PK3NETInitialize(net, "Project1.exe", Network, NetworkLen)
    ret = PK3NETStartNet
End Sub
```

■終了処理

```
Private Sub Form_Unload(Cancel As Integer)
    PK3NETStopNet
End Sub
```

■他 PC のアプリケーション実行監視イベント

```
Private Sub Form_Resize()
    Dim ret As Long
    Dim msg As PROTECTKIT3NET_MSG
    ' 通知メッセージ
    ret = PK3NETGetMessage(msg)
    If ret <> False Then
        Text2.Text = Left(msg.szMessage, msg.lMessageLen) ' 通知メッセージの取得
        Text4.Text = Left(msg.szPcName, msg.lPcNameLen)
        Text5.Text = Left(msg.szMacAdr, msg.lMacAdrLen)
        Beep
    End If
End Sub
```

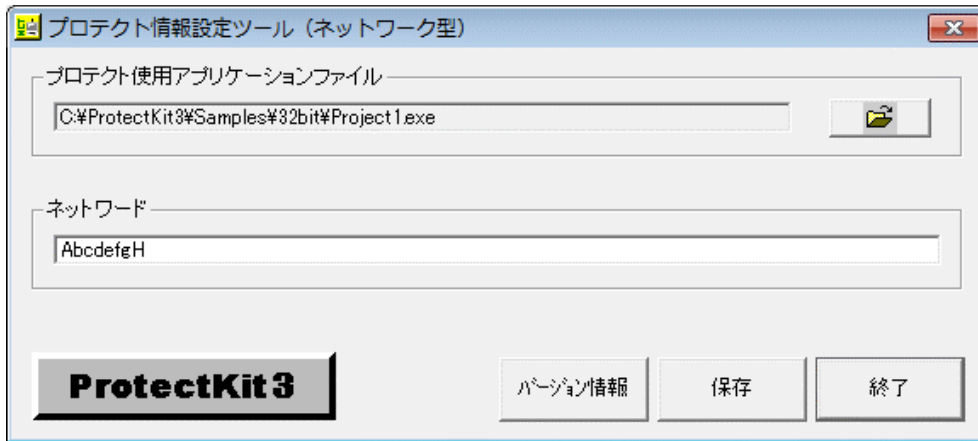
※詳しくは付属のサンプルプログラムをご覧ください。

STEP-2 ネットワードの設定

[プロテクト設定情報ツール(ネットワーク型)の利用]

作成したアプリケーションファイルにネットワードを設定します。

(アプリケーションを再構築した場合は再度ネットワードを設定する必要があります。)



※ネットワークは外部に流出しないようにお取り扱いに十分ご注意ください。

STEP-3 デジタル署名の添付

作成したアプリケーションファイルにデジタル署名を添付します。デジタル署名が不要な場合は STEP-4 へ。

STEP-4 アプリケーションの配布

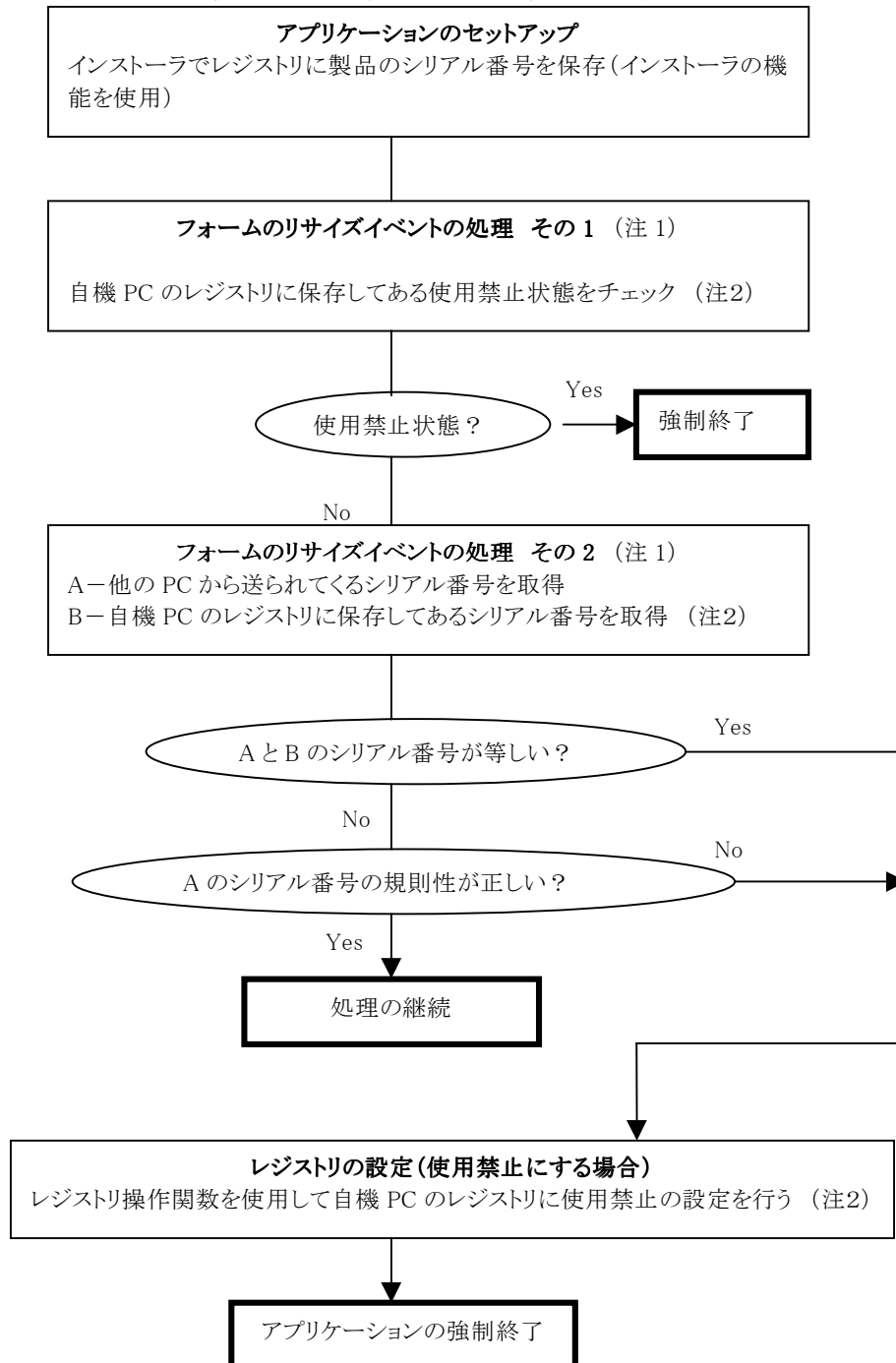
開発環境付属(または市販)のインストーラを利用し完成したアプリケーションと関連ファイルを配布します。エンドユーザへのパスワードの発行・通知は必要ありません。

※配布ファイルについては、「9. 作成したアプリケーションの配布」をご覧ください。

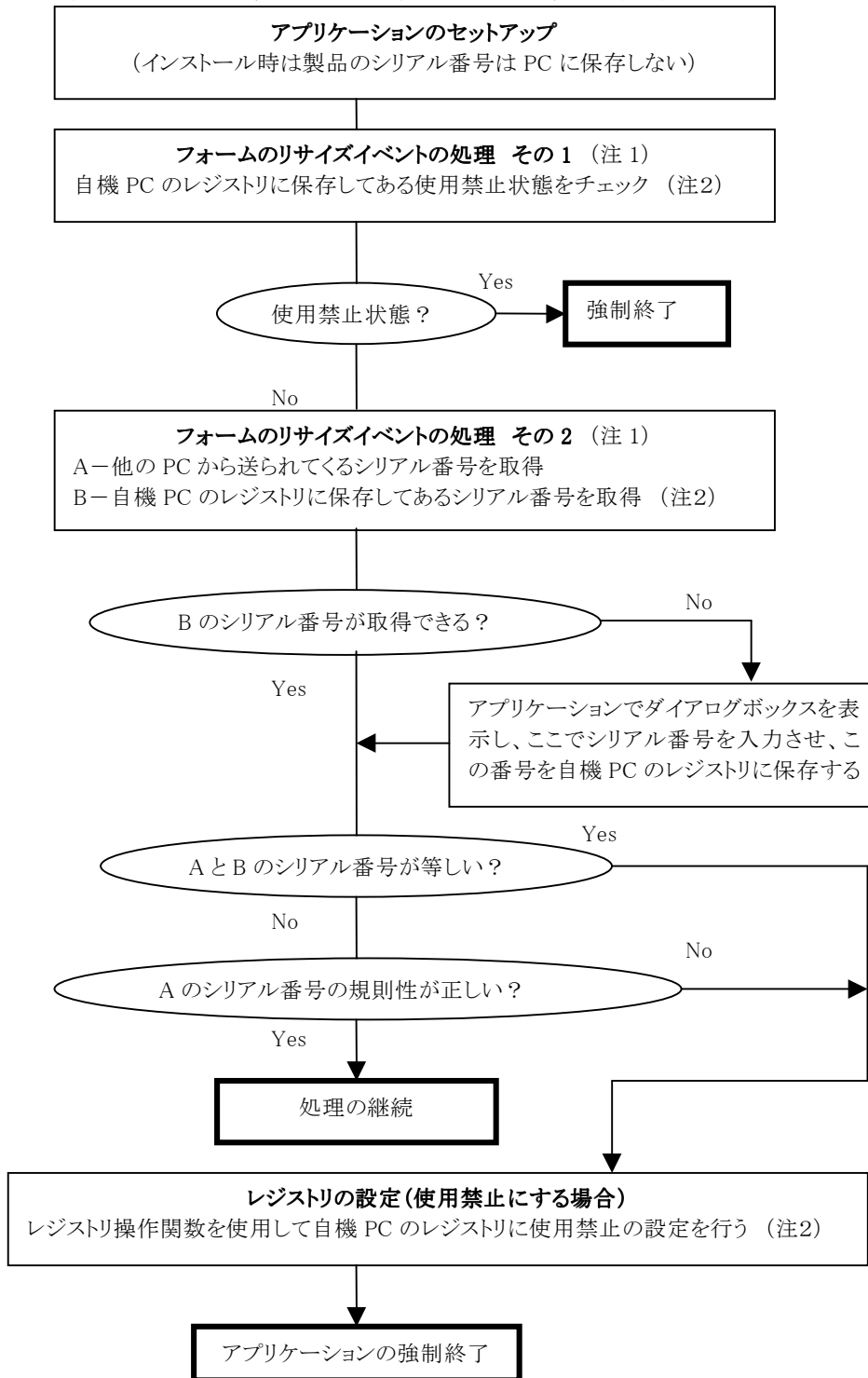
ネットワーク型プロテクトの運用例

以降の各ケースでは、LAN 上の自機 PC と他の PC を識別するためにアプリケーションのシリアル番号を用いています。つまり配布するアプリケーションには何らかの形(メディア、マニュアル、ユーザ登録カードへの添付など)で異なるシリアル番号をエンドユーザに提供する必要があります。

●ケース 1(インストーラで製品のシリアル番号を PC に保存する場合)



●ケース2(アプリケーションで製品のシリアル番号を PC に保存する場合)



プロテクトアプリケーションの運用手順

(注 1) フォームリサイズイベント

C++Builder

```
void _fastcall TForm1::FormResize(TObject *Sender)
```

Delphi

```
procedure TForm1.FormResize(Sender: TObject);
```

VB

```
Private Sub Form_Resize()
```

VC++

```
void CProject1Dlg::OnSize(UINT nType, int cx, int cy)
```

(注 2) レジストリ操作関数

値の取得 – PK3NETGetRegistryValue()

値の設定 – PK3NETSetRegistryValue()

削除 – PK3NETDeleteRegistryKey()

※DLL 関数については **ProtectKit3.chm** の「PkitNet32.dll/PkitNet64.dll」をご覧ください。

6. プロテクト情報設定ツールについて

6-1. パスワード発行型プロテクトの場合(ネイティブアプリケーション)

パスワード発行型プロテクトでお客様が作成したアプリケーションファイル(EXE)とプロテクト情報保持ファイルにプロテクト情報を設定します。(アプリケーションを再構築した場合は、再度プロテクト情報を設定する必要があります。)



【機能】

作成したアプリケーションファイル(プロテクト未完了アプリケーション)とプロテクト情報保持ファイルにプロテクト情報を設定します。プロテクト情報保持ファイルのファイル名は、ProtectKit3.5をインストールした状態(デフォルト)ではProInfo32.dll/ProInfo64.dllです。32ビットアプリケーションではProInfo32.dllを、64ビットアプリケーションではProInfo64.dllを使用します。

このファイルは下図のように、プロテクト完了アプリケーションの完成と同時にプロテクト情報が付加されます。

■プロテクト情報設定ツールを実行する前の状態

作成したアプリケーション
(プロテクト未設定アプリケーション)



プロテクト情報保持ファイル
(プロテクト未設定状態)



■プロテクト情報設定ツールを実行した後の状態

作成したアプリケーション
(プロテクト完了アプリケーション)



プロテクト情報保持ファイル
(プロテクト設定状態)



プロテクト未設定アプリケーションとは、開発コンテナで EXE などを作成した状態を言い、プロテクト完了アプリケーションは、プロテクト未設定アプリケーションに対してプロテクト情報を保存した状態(プロテクト情報設定ツールを実行した状態)を言います。プロテクト完了アプリケーションは元のアプリケーションのサイズより約 7KB 程度サイズが大きくなります。このように配布するプロテクトアプリケーションはプロテクト保持ファイルとのキーペアファイルとして動作することになります。

【起動方法】

プロテクト情報設定ツールについて

「プロテクト情報設定ツール」のアイコンや、**ProtectKit3.5** をインストール時に指定したフォルダの下の ¥ProtectKit3Pas¥SetProtect から「**Pr3SetProtect.exe**」を起動します。

【操作手順】

- (1)[開く]ボタンを使用してプロテクト情報保持ファイルを選択します。(デフォルトでは ProInfo32.dll/ProInfo64.dll)
- (2)[プロテクト使用アプリケーションファイル]のパネルにあるボタン(ファイルを開くアイコン)を使用して作成したアプリケーションを選択します。
- (3)[PC 識別条件]と[プロテクト条件]を設定します。
- (4)[上書き保存]ボタンもしくは[名前を付けて保存]ボタンを選択し、プロテクト情報保持ファイルと作成したアプリケーションに設定したプロテクト情報を保存します。
この操作で対象アプリケーションとプロテクト情報保持ファイルが更新され、配布可能なアプリケーションとプロテクト設定済のプロテクト情報保持ファイルが完成します。

【操作説明】

■ ボタン

- ・[開く]ボタン
プロテクト情報保持ファイルをオープンします。
- ・[上書き保存]ボタン
[開く]ボタンでオープンしたプロテクト情報保持ファイルにプロテクト情報を上書き保存します。
- ・[名前を付けて保存]ボタン
[開く]ボタンでオープンしたプロテクト情報保持ファイルとは別のファイル(同じでも可)に、プロテクト情報を保存します。ここでデフォルトのファイル名称 ProInfo32.dll/ProInfo64.dll を任意のファイル名にリネーム保存し、配布することで ProtectKit3 を使用したアプリケーションであることを推測させないために役立ちます。
- ・[新規]ボタン
[開く]ボタンでオープンしたプロテクト情報保持ファイルのプロテクト情報をクリアします。
- ・[終了]ボタン
プロテクト情報設定ツールを終了します。
- ・「プロテクト使用アプリケーションファイル」のパネルにあるボタン
作成したアプリケーションファイルをオープンします。

■ メニュー

- ・[ファイル(F)]メニュー
 - 開く(O) [開く]ボタンと同じ機能。
 - 上書き保存(S) [上書き保存]ボタンと同じ機能。
 - 名前を付けて保存(A) [名前を付けて保存]ボタンと同じ機能。
 - 新規(N) [新規]ボタンと同じ機能。
 - 終了(X) [終了]ボタンと同じ機能。
- ・[ヘルプ(H)]メニュー
 - 目次(C) ヘルプファイルを起動します。
 - バージョン情報(A) プロテクト情報設定ツールのバージョンを表示します。

■ PC 識別条件

- ・CPU・マザーボード・ビデオカード・ネットワークカードの全てが違う場合
4つの条件が全て違う場合に別のPCと判断します。
- ・下記の設定項目が一つでも違う場合
CPU、マザーボードのノースブリッジ、ビデオカード、ネットワークカード
指定した項目が一つでも違う場合に別のPCと判断します。
- ・PC 識別条件に乱数情報を付加
ハードウェアの条件が同じ時や、一定期間でパスワードを変更したい場合に設定します。
乱数情報を有効にすると、システムディスクのフォーマットやOSの再インストール時に新しいパスワードが必要となる反面、「プロテクト条件」の「パスワードの期限あり(乱数情報を付加の時)」を設定することにより、ソフトウェアのリース販売が行えます。
例えば、「プロテクト条件」の「使用月数」を1ヶ月と設定するとパスワードの有効期間が1ヶ月となり、次の月には別の新しいパスワードが必要となります。
また、乱数の保存場所となるレジストリの設定が必要となります。

PC 識別条件の文字列

PC 識別条件として PC 固有の文字列が設定に応じて生成されます。
例えば、

CPU

特性:10386100640

1-メーカー番号

0-3DNow 命令セット非対応(1-対応)

3-ステッピング ID

8-モデル

6-命令ファミリー

1-MMX 対応(0-非対応)

マザーボードのノースブリッジ

ベンダ番号:8086

製品番号:7124

ビデオカード

ベンダ番号:8086

製品番号:7125

ネットワークカード

MAC アドレス:00E0297CCF83

乱数

2529609067

の時、PC 固有の文字列は

10386100640808671248086712500E0297CCF832529609067

となります

(注意)

- 1.エンデューザは、CPU やビデオカード及びネットワークカードを交換する場合があります。
- 2.ビデオカードの情報が取得できない PC が存在する場合があります。
- 3.PC 識別条件に CPU またはネットワークカードを指定し、複数の CPU やネットワークカードが搭載された PC で運用すると、CPU 情報とネットワークカードの情報が不定になります。

■プロテクト条件

- ・ロックワード(英数字 8 文字以上)

パスワードの基となるロックワード(鍵文字列)を設定します。

半角の英数字で 8~64 文字以内で設定してください。

ロックワードはアプリケーション毎に設定してください。

(注意)ロックワードは絶対に外部に漏らさないでください。

- ・プロテクト方法

1.標準プロテクト パスワードの発行を受けない限り使用できない

2.使用回数でプロテクト 設定回数分の使用が可能

3.使用日数でプロテクト 設定日数分の使用が可能

4.使用月数でプロテクト 設定月数分の使用が可能

5.使用期限でプロテクト 設定期限までの使用が可能

- ・使用回数

「使用回数でプロテクト」をかける時にパスワードを発行せずにアプリケーションを使用できる回数を設定します。

- ・使用日数

「使用日数でプロテクト」をかける時にパスワードを発行せずにアプリケーションを使用できる日数を設定します。

- ・使用月数

「使用月数でプロテクト」をかける時にパスワードを発行せずにアプリケーションを使用できる月数を設定します。

その際に開始日を 3 種類の中から選択できます。

開始日

0:最初の実行日から

1:最初の実行日の当月の初めから

2:最初の実行日の翌月の初めから

(例)使用月数が 2 ヶ月で 2001/03/16 にアプリケーションを最初に実行した場合

0:2001/03/16 から使用月数のプロテクトを開始し、2001/05/15 までは使用可

1:2001/03/01 から使用月数のプロテクトを開始し、2001/04/30 までは使用可

2:2001/04/01 から使用月数のプロテクトを開始し、2001/05/31 までは使用可

- ・使用期限

プロテクト情報設定ツールについて

「使用期限でプロテクト」をかける時、パスワードを発行せずにアプリケーションを使用できる期限を設定します。

・パスワードの期限あり(乱数情報を付加の時)

一定期間毎にパスワードを変更する場合に選択します。

設定が有効になるのは、プロテクト方法が「使用回数でプロテクト」、「使用日数でプロテクト」、「使用月数でプロテクト」となります。

(1)設定なしの場合

使用回数や使用日数もしくは使用月数の残りがなくなるまでアプリケーションを使用できます。

(2)設定ありの場合

アプリケーションを最初に起動するとプロテクト状態となりますので、プロテクトを解除するためのパスワードが必要になります。パスワードを発行すると設定した使用回数や使用日数もしくは使用月数の残りがなくなるまでアプリケーションを使用できます。残りがなくなった後で延長して使用する場合は再度別のパスワードの発行が必要になります。(この流れを延々と繰り返すこととなります。)

※有効期限を過ぎると有効期限内の PciID とは異なる PciID が返されます。

・パスワードの構成

1.数字

数字のみのパスワードとなります。(例 12345678)

2.数字 + 英字

数字と英字(大小文字)の組み合わせのパスワードとなります。(例 12345678ABCabc)

3.数字 + 英字(大文字)

数字と英大文字の組み合わせのパスワードとなります。(例 12345678ABC)

4.ProtectKit2.x のパスワードとの互換性

PC 識別条件に乱数情報を付加しない場合(乱数情報が無効)に選択可能です。

有効にすると、ProtectKit2/2.5 で作成したアプリケーションと同じパスワードが利用できます。ただし、ロックワード・PC 識別条件・パスワードの構成が同じ場合が対象です。

■レジストリ、ファイル

使用回数や使用日数、乱数などのプロテクト情報をレジストリとファイルの両方に保存することができます。仮にエンドユーザがプロテクト情報を保存しているレジストリを削除してもファイルが残っていればレジストリを復旧し、逆にファイルを削除してもレジストリが残っていればファイルを復旧することができます。ただし、レジストリとファイルの両方を削除された場合には復旧することができません。

A. レジストリ

・キー

プロテクト情報の保存場所となるレジストリのメインキーを設定します。

HKEY_CLASSES_ROOT

HKEY_LOCAL_MACHINE\SOFTWARE

・プロテクト状態の保存場所

プロテクト状態の保存場所を設定します。

ただし、**プロテクト方法が標準プロテクト以外の場合**に有効となります。

サブキー

レジストリのサブキーを設定します。

名前

値を保存するキーの名前を設定します。

・乱数情報の保存場所

乱数情報の保存場所を設定します。

ただし、「**PC 識別条件に乱数情報を付加**」を設定した場合に有効となります。

サブキー

レジストリのサブキーを設定します。

名前

値を保存するキーの名前を設定します。

(注意)レジストリの保存場所は、なるべく簡単にわからないように深い階層に設定し、サブキーと名前はアプリケ

ーション名から想像出来ない名称に設定してください。

B. ファイル

・フォルダ

プロテクト情報の保存場所となるフォルダを設定します。

Windows フォルダ

Windows の System フォルダ

(Windows XP では一般的に"C:¥Windows"または"C:¥Windows¥system32"となります)

・プロテクト状態の保存場所

プロテクト状態の保存場所を設定します。

ただし、**プロテクト方法が標準プロテクト以外の場合**に有効となります。

サブフォルダ

上記のフォルダ直下のサブフォルダを設定します。

(注意)

実際に存在するフォルダ名を設定してください。サブフォルダを設定しない場合、上記フォルダ直下に下記のファイル名のファイルが作成されます。

ファイル名

値を保存するキーの名前を設定します。

・乱数情報の保存場所

乱数情報の保存場所を設定します。

ただし、「**PC 識別条件に乱数情報を付加**」を設定した場合に有効となります。

サブフォルダ

上記のフォルダ直下のサブフォルダを設定します。

(注意)

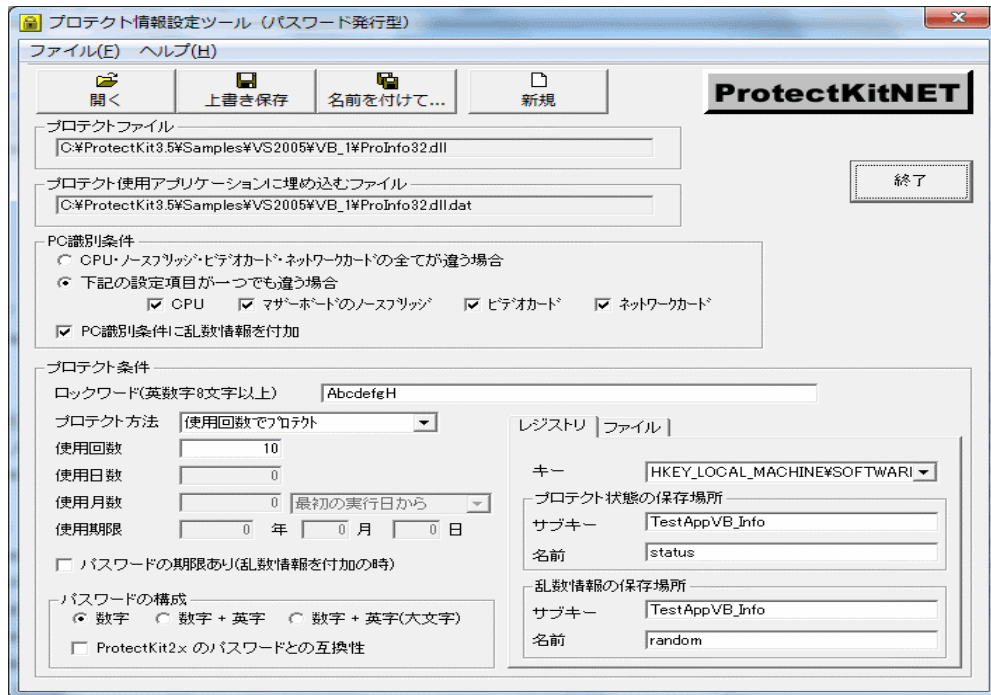
実際に存在するフォルダ名を設定してください。サブフォルダを設定しない場合、上記フォルダ直下に下記のファイル名のファイルが作成されます。

ファイル名

値を保存するキーの名前を設定します。

6-2. パスワード発行型プロテクトの場合 (CLR アプリケーション)

パスワード発行型プロテクトでプロテクト情報保持ファイルとアプリケーションに埋め込むファイルにプロテクト情報を設定します。



【機能】

プロテクト情報保持ファイルとアプリケーションに埋め込むファイルにプロテクト情報を設定します。
 プロテクト情報保持ファイルのファイル名は、ProtectKit3.5 をインストールした状態 (デフォルト) では ProInfo32.dll/ProInfo64.dll です。**32 ビットアプリケーションでは ProInfo32.dll を、64 ビットアプリケーションでは ProInfo64.dll を使用します。**

【起動方法】

「プロテクト情報設定ツール」のアイコンや、ProtectKit3.5 をインストール時に指定したフォルダの下の ¥ProtectKit3Pas¥SetProtect から「PrNETSetProtect.exe」を起動します。

【操作手順】

- (1)[開く]ボタンを使用してプロテクト情報保持ファイルを選択します。(デフォルトでは ProInfo32.dll/ProInfo64.dll)
- (2)[PC 識別条件]と[プロテクト条件]を設定します。
- (3)[上書き保存]ボタンもしくは[名前を付けて保存]ボタンを選択し、プロテクト情報保持ファイルとプロテクト使用アプリケーションに埋め込むファイルに設定したプロテクト情報を保存します。
 この操作の後で、プロテクト使用アプリケーションに埋め込むファイルを対象となるプロジェクトに追加してビルドすることで配布可能なアプリケーションとプロテクト設定済のプロテクト情報保持ファイルが完成します。

【操作説明】

■ ボタン

- ・[開く]ボタン
 プロテクト情報保持ファイルをオープンします。
- ・[上書き保存]ボタン
 [開く]ボタンでオープンしたプロテクト情報保持ファイルにプロテクト情報を上書き保存します。
- ・[名前を付けて保存]ボタン
 [開く]ボタンでオープンしたプロテクト情報保持ファイルとは別のファイル (同じでも可) に、プロテクト情報を保存します。ここでデフォルトのファイル名称 ProInfo32.dll/ProInfo64.dll を任意のファイル名にリネーム保存保存することができます。
- ・[新規]ボタン
 [開く]ボタンでオープンしたプロテクト情報保持ファイルのプロテクト情報をクリアします。
- ・[終了]ボタン
 プロテクト情報設定ツールを終了します。

■メニュー

- ・[ファイル(F)]メニュー
 - 開く(O) [開く]ボタンと同じ機能。
 - 上書き保存(S) [上書き保存]ボタンと同じ機能。
 - 名前を付けて保存(A) [名前を付けて保存]ボタンと同じ機能。
 - 新規(N) [新規]ボタンと同じ機能。
 - 終了(X) [終了]ボタンと同じ機能。
- ・[ヘルプ(H)]メニュー
 - 目次(C) ヘルプファイルを起動します。
 - バージョン情報(A) プロテクト情報設定ツールのバージョンを表示します。

■PC 識別条件

- ・CPU・マザーボード・ビデオカード・ネットワークカードの全てが違う場合
4つの条件が全て違う場合に別のPCと判断します。
- ・下記の設定項目が一つでも違う場合
CPU、マザーボードのノースブリッジ、ビデオカード、ネットワークカード
指定した項目が一つでも違う場合に別のPCと判断します。
- ・PC 識別条件に乱数情報を付加
ハードウェアの条件が同じ時や、一定期間でパスワードを変更したい場合に設定します。
乱数情報を有効にすると、システムディスクのフォーマットやOSの再インストール時に新しいパスワードが必要となる反面、「プロテクト条件」の「パスワードの期限あり(乱数情報を付加の時)」を設定することにより、ソフトウェアのリース販売が行えます。
例えば、「プロテクト条件」の「使用月数」を1ヶ月と設定するとパスワードの有効期間が1ヶ月となり、次の月には別の新しいパスワードが必要となります。
また、乱数の保存場所となるレジストリの設定が必要となります。

PC 識別条件の文字列

PC 識別条件としてPC固有の文字列が設定に応じて生成されます。

例えば、

```

CPU
  特性: 10386100640
    1-メーカー番号
    0-3DNow 命令セット非対応(1-対応)
    3-ステッピング ID
    8-モデル
    6-命令ファミリー
    1-MMX 対応(0-非対応)
マザーボードのノースブリッジ
  ベンダ番号: 8086
  製品番号: 7124
ビデオカード
  ベンダ番号: 8086
  製品番号: 7125
ネットワークカード
  MAC アドレス: 00E0297CCF83
乱数
  2529609067
の時、PC 固有の文字列は
  10386100640808671248086712500E0297CCF832529609067
となります
  
```

(注意)

1. エンドユーザは、CPU やビデオカード及びネットワークカードを交換する場合があります。
2. ビデオカードの情報が取得できないPCが存在する場合があります。
3. PC 識別条件にCPUまたはネットワークカードを指定し、複数のCPUやネットワークカードが搭載されたPCで運用すると、CPU情報とネットワークカードの情報が不定になります。

■プロテクト条件

- ・ロックワード(英数字8文字以上)
パスワードの基となるロックワード(鍵文字列)を設定します。

プロテクト情報設定ツールについて

半角の英数字で 8～64 文字以内で設定してください。
パスワードはアプリケーション毎に設定してください。
(注意)パスワードは絶対に外部に漏らさないでください。

・プロテクト方法

- 1.標準プロテクト パスワードの発行を受けない限り使用できない
- 2.使用回数でプロテクト 設定回数分の使用が可能
- 3.使用日数でプロテクト 設定日数分の使用が可能
- 4.使用月数でプロテクト 設定月数分の使用が可能
- 5.使用期限でプロテクト 設定期限までの使用が可能

・使用回数

「使用回数でプロテクト」をかける時にパスワードを発行せずにアプリケーションを使用できる回数を設定します。

・使用日数

「使用日数でプロテクト」をかける時にパスワードを発行せずにアプリケーションを使用できる日数を設定します。

・使用月数

「使用月数でプロテクト」をかける時にパスワードを発行せずにアプリケーションを使用できる月数を設定します。
その際に開始日を 3 種類の中から選択できます。

開始日

- 0:最初の実行日から
- 1:最初の実行日の当月の初めから
- 2:最初の実行日の翌月の初めから

(例)使用月数が 2 ヶ月で 2001/03/16 にアプリケーションを最初に実行した場合

- 0:2001/03/16 から使用月数のプロテクトを開始し、2001/05/15 までは使用可
- 1:2001/03/01 から使用月数のプロテクトを開始し、2001/04/30 までは使用可
- 2:2001/04/01 から使用月数のプロテクトを開始し、2001/05/31 までは使用可

・使用期限

「使用期限でプロテクト」をかける時、パスワードを発行せずにアプリケーションを使用できる期限を設定します。

・パスワードの期限あり(乱数情報を付加の時)

一定期間毎にパスワードを変更する場合に選択します。

設定が有効になるのは、プロテクト方法が「使用回数でプロテクト」、「使用日数でプロテクト」、「使用月数でプロテクト」となります。

(1)設定なしの場合

使用回数や使用日数もしくは使用月数の残りがなくなるまでアプリケーションを使用できます。

(2)設定ありの場合

アプリケーションを最初に起動するとプロテクト状態となりますので、プロテクトを解除するためのパスワードが必要になります。パスワードを発行すると設定した使用回数や使用日数もしくは使用月数の残りがなくなるまでアプリケーションを使用できます。残りがなくなった後で延長して使用する場合は再度別のパスワードの発行が必要になります。(この流れを延々と繰り返すこととなります。)

※有効期限を過ぎると有効期限内の PcID とは異なる PcID が返されます。

・パスワードの構成

1.数字

数字のみのパスワードとなります。(例 12345678)

2.数字 + 英字

数字と英字(大小文字)の組み合わせのパスワードとなります。(例 12345678ABCabc)

3.数字 + 英字(大文字)

数字と英大文字の組み合わせのパスワードとなります。(例 12345678ABC)

4.ProtectKit2.x のパスワードとの互換性

PC 識別条件に乱数情報を付加しない場合(乱数情報が無効)に選択可能です。

有効にすると、ProtectKit2/2.5 で作成したアプリケーションと同じパスワードが利用できます。ただし、パスワード・PC 識別条件・パスワードの構成が同じ場合が対象です。

■レジストリ、ファイル

使用回数や使用日数、乱数などのプロテクト情報をレジストリとファイルの両方に保存することができます。仮にエンドユーザがプロテクト情報を保存しているレジストリを削除してもファイルが残っていればレジストリを復旧し、逆にファイルを削除してもレジストリが残っていればファイルを復旧することができます。ただし、レジストリとファイルの両方を削除された場合には復旧することができません。

A. レジストリ

・キー

プロテクト情報の保存場所となるレジストリのメインキーを設定します。

HKEY_CLASSES_ROOT
HKEY_LOCAL_MACHINE¥SOFTWARE

・プロテクト状態の保存場所

プロテクト状態の保存場所を設定します。

ただし、**プロテクト方法が標準プロテクト以外の場合**に有効となります。

サブキー

レジストリのサブキーを設定します。

名前

値を保存するキーの名前を設定します。

・乱数情報の保存場所

乱数情報の保存場所を設定します。

ただし、「**PC 識別条件に乱数情報を付加**」を設定した場合に有効となります。

サブキー

レジストリのサブキーを設定します。

名前

値を保存するキーの名前を設定します。

(注意)レジストリの保存場所は、なるべく簡単にわからないように深い階層に設定し、サブキーと名前はアプリケーション名から想像出来ない名称に設定してください。

B. ファイル

・フォルダ

プロテクト情報の保存場所となるフォルダを設定します。

Windows フォルダ
Windows の System フォルダ
(Windows XP では一般的に"C:¥Windows"または"C:¥Windows¥system32"となります)

・プロテクト状態の保存場所

プロテクト状態の保存場所を設定します。

ただし、**プロテクト方法が標準プロテクト以外の場合**に有効となります。

サブフォルダ

上記のフォルダ直下のサブフォルダを設定します。

(注意)

実際に存在するフォルダ名を設定してください。サブフォルダを設定しない場合、上記フォルダ直下に下記のファイル名のファイルが作成されます。

ファイル名

値を保存するキーの名前を設定します。

・乱数情報の保存場所

乱数情報の保存場所を設定します。

ただし、「**PC 識別条件に乱数情報を付加**」を設定した場合に有効となります。

サブフォルダ

上記のフォルダ直下のサブフォルダを設定します。

(注意)

実際に存在するフォルダ名を設定してください。サブフォルダを設定しない場合、上記フォルダ直下に下記のファイル名のファイルが作成されます。

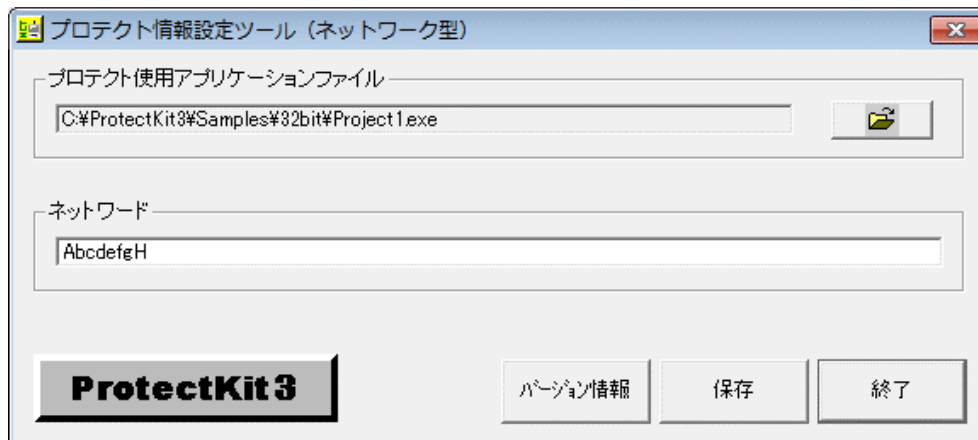
プロテクト情報設定ツールについて

ファイル名

値を保存するキーの名前を設定します。

6-3. ネットワーク型プロテクトの場合(ネイティブアプリケーション)

ネットワーク型プロテクトでお客様が作成したアプリケーションファイル(EXE)にパスワードを設定します。(アプリケーションを再構築した場合は、再度パスワードを設定する必要があります。)

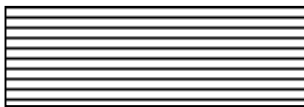


【機能】

作成したアプリケーションファイル(プロテクト未完了アプリケーション)にパスワードを設定します。ネットワーク型プロテクトは、パスワード発行型プロテクトの場合とは異なり、プロテクト情報保持ファイル(デフォルトでは ProInfo32.dll/ProInfo64.dll)は使用しません。

■プロテクト情報設定ツールを実行する前の状態

作成したアプリケーション
(プロテクト未設定アプリケーション)



■プロテクト情報設定ツールを実行した後の状態

作成したアプリケーション
(プロテクト完了アプリケーション)



プロテクト未設定アプリケーションとは、開発コンテナで EXE などを作成した状態を言い、プロテクト完了アプリケーションは、プロテクト未設定アプリケーションに対してパスワードを保存した状態(プロテクト情報設定ツールを実行した状態)を言います。プロテクト完了アプリケーションは元のアプリケーションのサイズより約 7KB 程度サイズが大きくなります。

【起動方法】

「プロテクト情報設定ツール」のアイコンや、**ProtectKit3.5** をインストール時に指定したフォルダの下の ¥ProtectKit3Net¥SetProtectNet から「Pr3SetProtectNET.exe」を起動します。

【操作手順】

- (1)[ファイルを開くアイコン]ボタンを使用して作成したアプリケーションを選択します。
- (2)[保存]ボタンを選択し、作成したアプリケーションに設定したパスワードを保存します。

プロテクト情報設定ツールについて

- [バージョン情報]ボタン
プロテクト情報設定ツールのバージョンを表示します。
- [保存]ボタン
作成したアプリケーションに設定したネットワークを保存します。
- [終了]ボタン
プロテクト情報設定ツールを終了します。

■ネットワーク

半角の英数字で 8～64 文字以内で設定してください。
ネットワークはアプリケーション毎に設定してください。
(注意) ネットワークは絶対に外部に漏らさないでください。

7. パスワード発行について

ユーザから通知された PcID をもとにパスワードを発行いたします。

パスワードは 3 通りの方法で発行できます。(EXE・DLL・OCX)

いずれの場合も、取得したパスワードをユーザへ通知し、ユーザがそれを入力することでアプリケーションのプロテクトを解除します。

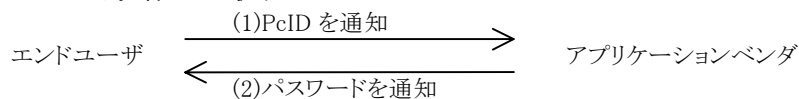
実行型プログラム(EXE)では、ロックワードとユーザから通知された PcID の二つの情報を入力することにより、パスワードを発行する仕組みになっております。

DLL と OCX でも同様な機能になりますので、お客様サイトでカスタマイズしたパスワード発行ツールを作成する場合などにご利用いただけます。

プロテクト情報設定ツールの「パスワードの構成 - ProtectKit2.x のパスワードとの互換性」を有効にする場合は ProtectKit2/2.5 付属のパスワード発行ツール(もしくは OCX や DLL)を使用してください。

一方、無効(デフォルト時)の場合は、ProtectKit3/3.5 付属のパスワード発行ツール(もしくは OCX や DLL)を使用してください。

パスワード発行までの流れ



A. パスワードの有効期限なしの場合

一度パスワードを発行すれば、それ以後エンドユーザとの間で PcID とパスワードのやり取りは必要ありません。

B. パスワードの有効期限ありの場合

最初にエンドユーザとの間で PcID とパスワードのやり取りを行い、有効期限が過ぎた後でまた PcID とパスワードのやり取りを行う必要があります。(有効期限が過ぎると PcID が変わるためです。)

※「パスワードの期限あり」の設定は、プロテクト情報設定ツール(パスワード発行型)で行います。

詳しくは「7. プロテクト情報設定ツールについて」をご覧ください。

7-1. 付属の実行型プログラムを使用する方法(パスワード発行ツール)

Pr3PassGen.exe



【機能】

プロテクトを解除するためのパスワードを発行します。

【起動方法】

「パスワード発行ツール」のアイコンや、**ProtectKit3.5** をインストール時に指定したフォルダの下の ¥ProtectKit3Pas¥PasswordGenerator¥EXE から「**Pr3PassGen.exe**」を起動します。

【操作手順】

- (1)[ロックワード]ボックスにアプリケーションに設定したロックワードを入力します。
- (2)[ユーザマシンの PC 情報(PcID)]ボックスにユーザから通知された「PcID」を入力します。
- (3)[パスワードの構成]を選択し、[パスワードの取得]ボタンを押すと[パスワード]ボックスにプロテクトを解除するためのパスワードが表示されます。

パスワード文字列の識別支援機能

[パスワード]ボックスの下には、現在カーソル位置の文字の「読み」が表示され、右下にはその文字の属性が示されます。この機能は、l(小文字のエル)とI(大文字のアイ)などのように識別しにくい文字を明確にすることができます。

- [バージョン情報]ボタン
パスワード発行ツールのバージョンを表示します。
- [フォント設定]ボタン
[ロックワード]、[ユーザマシンの PC 情報(PcID)]、[パスワード]の各ボックスに表示するフォントを設定します。
- [OK]ボタン
入力したロックワードとフォント情報が保存され、次回起動時に設定されます。

7-2. 付属の DLL を使用する方法(パスワード発行 DLL)

Pr3PassGen.dll(32 ビットアプリケーション用)

- Pr3PassGen.dll のコマンドを使用する前に
呼び出し側のプログラムで DLL コマンド(プロシージャ)のプロトタイプ宣言の記述が必要になります。

コンテナ	ヘッダファイル名
C++Builder、VisualC++	Pr3PassGen.h
Delphi	Pr3PassGen.pas
VisualBasic	Pr3PassGen.bas

- DLL 関数コマンド一覧(アルファベット順)

関数名	内容
PS3GetPassword	パスワードの発行
PS3SendMail	Eメールの送信

PS3GetPassword

【機能】

パスワードを発行します。

【関数書式】

(1)C++Builder,VisualC++

```
BOOL PS3GetPassword(LPCSTR lpszLockword, LPCSTR lpszPcID, long l PasswordType, LPSTR lpszPassword);
```

(2)Delphi

```
function PS3GetPassword(lpszLockword, lpszPcID: PAnsiChar; lPasswordType: Longint; lpszPassword: PAnsiChar): LongBool;
```

(3)VisualBasic

```
Function PS3GetPassword(ByVal lpszLockword As String, ByVal lpszPcID As String, ByVal lPasswordType As Long, ByVal lpszPassword As String) As Long
```

【引数】

名称	内容
lpszLockword	アプリケーションに設定したロックワード
lpszPcID	ユーザから通知された PcID
lPasswordType	パスワードの構成 0:数字のみ 1:数字+アルファベットの大小文字の組み合わせ 2:数字+アルファベットと大文字の組み合わせ
lpszPassword	取得するパスワード (C++Builder,VisualC++,Delphi はヌルで終了する文字列を与え、VisualBasic はサイズを指定した String 型を与えること)

【戻り値】

実行に失敗した場合は False(0)、成功した場合は True(0 以外)が返されます。

PS3SendMail

【機能】

Eメールの送信を行います。

【関数書式】

(1)C++Builder,VisualC++

```
long PS3SendMail(LPCSTR lpszMailSvrAddr, LPCSTR lpszFromAddr, LPCSTR lpszFromHdr, LPCSTR lpszToAddr, LPCSTR lpszToHdr, LPCSTR lpszSubject, LPCSTR lpszMessage);
```

(2)Delphi

```
function PS3SendMail(lpszMailSvrAddr, lpszFromAddr, lpszFromHdr, lpszToAddr, lpszToHdr, lpszSubject, lpszMessage: PAnsiChar): Longint;
```

(3)VisualBasic

```
Function PS3SendMail(ByVal lpszMailSvrAddr As String, ByVal lpszFromAddr As String, ByVal lpszFromHdr As String, ByVal lpszToAddr As String, ByVal lpszToHdr As String, ByVal lpszSubject As String, ByVal lpszMessage As String) As Long
```

【引数】

名称	内容
lpszMailSvrAddr	送信(SMTP)メールサーバーのアドレス
lpszFromAddr	差出メールアドレス
lpszFromHdr	差出名
lpszToAddr	宛先メールアドレス
lpszToHdr	宛先名
lpszSubject	件名
lpszMessage	送信する内容

【戻り値】

	内容
0	正常
1	バージョン 1.1 以上の Winsock がありません
2	SMTP メールサーバーが見つかりません
3	メールサーバーのソケットをオープンできません
4	サーバーのソケットへの接続に失敗しました
5	引数が不正
6	メモリ確保に失敗
7	通信エラー

【解説】

送信する内容に改行を入れる場合は、文章の中に改行コードを入れてください。差出名・宛先名・件名・送信する内容に半角カタカナがある場合は全角カタカナに変換して送信します。
ダイヤルアップ接続の場合は、回線を接続した状態で使用してください。

7-3. 付属の OCX を使用する方法 (パスワード発行 OCX)

Pr3PassGen.ocx (32 ビットアプリケーション用)

●プロパティ一覧 (アルファベット順)

カスタムプロパティ	内容
szFromAddr	差出メールアドレス
szFromHdr	差出名
szLockword	アプリケーションに設定したロックワード
szMailSvrAddr	送信 (SMTP) メールサーバーのアドレス
szMessage	送信する内容
szPassword	取得するパスワード
lPasswordType	パスワードの構成
szPcID	ユーザから通知された PcID
szSubject	件名
szToAddr	宛先メールアドレス
szToHdr	宛先名

●メソッド一覧 (アルファベット順)

カスタムメソッド	内容
PS3GetPassword	パスワードの発行
PS3SendMail	Eメールの送信

**szFromAddr、szFromHdr、szMailSvrAddr、szToAddr、szToHdr、szSubject
(カスタムプロパティ)**
【機能】

Eメールを送信する際の差出及び宛先情報を設定します。

【書式】

※szFromAddrにて説明 (szFromHdr,szMailServerAddr,szToAddr,szToHdr,szSubjectも同様)

(1)C++Builder

controlname->szFromAddr [= WideString]

(2)Delphi

controlname.szFromAddr [= WideString]

(3)VisualBasic

controlname.szFromAddr [= String]

(4)VisualC++

controlname.SetszFromAddr(LPSTR Val)

【設定値】

szFromAddr プロパティは差出メールアドレス

szFromHdr プロパティは差出名

szMailSvrAddr プロパティは送信 (SMTP) メールサーバーのアドレス

szSubject プロパティは件名

szToAddr プロパティは宛先メールアドレス

szToHdr プロパティは宛先名

【解説】

PS3SendMail メソッドで活用されます。

【値の設定】 デザイン時、実行時

【値の参照】 実行時

szLockword (カスタムプロパティ)

【機能】

パスワードを発行する際に必要となるロックワードを設定します。

【書式】

- (1)C++Builder
`controlname->szLockword [= WideString]`
- (2)Delphi
`controlname.szLockword [= WideString]`
- (3)VisualBasic
`controlname.szLockword [= String]`
- (4)VisualC++
`controlname.SetszLockword(LPSTR Val)`

【設定値】

アプリケーションに設定したロックワード

【解説】

PS3GetPassword メソッドで活用されます。

【値の設定】 デザイン時、実行時

【値の参照】 実行時

szMessage (カスタムプロパティ)

【機能】

E メールを送信する際の内容 (文章) を設定します。

【書式】

- (1)C++Builder
`controlname->szMessage [= WideString]`
- (2)Delphi
`controlname.szMessage [= WideString]`
- (3)VisualBasic
`controlname.szMessage [= String]`
- (4)VisualC++
`controlname.SetszMessage(LPSTR Val)`

【設定値】

メールで送信する内容 (文章)

【解説】

送信する内容に改行を入れる場合は、文章の中に改行コードを入れてください。
PS3SendMail メソッドで活用されます。

【値の設定】 デザイン時、実行時

【値の参照】 実行時

szPassword(カスタムプロパティ)

【機能】

プロテクトを解除するパスワードを取得します。

【書式】

- (1)C++Builder
`controlname->szPassword [= WideString]`
- (2)Delphi
`controlname.szPassword [= WideString]`
- (3)VisualBasic
`controlname.szPassword [= String]`
- (4)VisualC++
`[LPSTR =]controlname.GetszPassword()`

【参照値】

パスワード

【解説】

PS3GetPassword メソッドが成功すると値が設定されます。

【値の設定】 不可

【値の参照】 実行時

IPasswordType (カスタムプロパティ)

【機能】

発行するパスワードの構成を設定します。

【書式】

- (1)C++Builder
`controlname->IPasswordType [= short]`
- (2)Delphi
`controlname.IPasswordType [= Smallint]`
- (3)VisualBasic
`controlname.IPasswordType [= Integer]`
- (4)VisualC++
`controlname.SetIPasswordType(short Val)`

【設定値】

- 0:数字のみ
- 1:数字+アルファベットの大小文字の組み合わせ
- 2:数字+アルファベット大文字の組み合わせ

【解説】

PS3GetPassword メソッドで活用されます。

【値の設定】 不可

【値の参照】 実行時

szPcID (カスタムプロパティ)

【機能】

パスワードを発行する際に必要となる PcID を設定します。

【書式】

- (1)C++Builder
controlname->szPcID [= WideString]
- (2)Delphi
controlname.szPcID [= WideString]
- (3)VisualBasic
controlname.szPcID [= String]
- (4)VisualC++
controlname.SetszPcID(LPSTR Val)

【設定値】

ユーザから通知された PcID

【解説】

PS3GetPassword メソッドで活用されます。

【値の設定】 デザイン時、実行時

【値の参照】 実行時

PS3GetPassword (カスタムメソッド)

【機能】

パスワードを発行します。

【書式】

- (1)C++Builder
[*BOOL* =] *controlname*->PS3GetPassword()
- (2)Delphi
[*Integer* =] *controlname*.PS3GetPassword
- (3)VisualBasic
[*Long* =] *controlname*.PS3GetPassword
- (4)VisualC++
[*BOOL* =] *controlname*.PS3GetPassword()

【引数】

ありません。

【戻り値】

実行に失敗した場合は False(0)、成功した場合は True(0 以外)が返されます。

【解説】

パスワードを発行するには **szLockword**、**szPcID**、**lPasswordType** プロパティを設定する必要があります。メソッドの実行に成功すると **szPassword** プロパティにパスワードが設定されます。

PS3SendMail (カスタムメソッド)

【機能】

Eメールの送信を行います。

(1)C++Builder

[*long* =] *controlname*->PS3SendMail()

(2)Delphi

[*Integer* =] *controlname*.PS3SendMail

(3)VisualBasic

[*Long* =] *controlname*.PS3SendMail

(4)VisualC++

[*long* =] *controlname*.PS3SendMail()

【引数】

ありません。

【戻り値】

内容

0	正常
1	バージョン 1.1 以上の Winsock がありません
2	SMTP メールサーバーが見つかりません
3	メールサーバーのソケットをオープンできません
4	サーバーのソケットへの接続に失敗しました
5	プロパティが不正
6	メモリ確保に失敗
7	通信エラー

【解説】

メソッドを実行する前に **szMailSvrAddr**、**szFromAddr**、**szFromHdr**、**szToAddr**、**szToHdr**、**szSubject**、**szMessage** プロパティを設定する必要があります。**szFromHdr**、**szToHdr**、**szSubject**、**szMessage** プロパティに半角カタカナがある場合は全角カタカナに変換して送信します。

ダイヤルアップ接続の場合は、回線を接続した状態で使用してください。

8. 運用上の注意事項

パスワード発行型プロテクトアプリケーションとネットワーク型プロテクトアプリケーションを運用する上での注意事項を説明します。

1. パスワード発行型プロテクト

■ ロックワードの外部流出に注意

プロテクト情報設定ツール(パスワード発行型)で設定したロックワードは外部に流出しないように十分注意してください。

■ プロテクト完了アプリケーション(EXE)とプロテクト情報保持ファイルの関係

一つのプロテクト完了アプリケーション(EXE)に対して一つのプロテクト情報保持ファイルが必要になります(一対一の関係)。

そのため、複数の実行プログラムにプロテクトをかけて配布するには、相当数の「プロテクト完了アプリケーション(EXE)」と複数の「プロテクト情報保持ファイル」が必要です。

■ DdScm.exe, DdScm32.exe, DdScm64.exe [プロテクト NT サービス]の登録と解除

ProtectKit3/ProtectKitNET を利用したアプリケーションを実行する場合は、アプリケーションの最初の起動前に、DdScm.exe, DdScm32.exe, DdScm64.exe [プロテクト NT サービス]を登録し、そのサービスが開始されている必要があります。DdScm.exe, DdScm32.exe, DdScm64.exe は Windows のユーザ権限に関わらずプロテクト機能を有効にするサービスです。

DdScm.exe の登録と解除を行う場合は、同じフォルダに DdScm32.exe または DdScm64.exe が必要です。

DdScm.exe の登録 DdScm.exe /a

DdScm.exe の解除 DdScm.exe /r

DdScm32.exe の登録 DdScm32.exe /a

DdScm32.exe の解除 DdScm32.exe /r

DdScm64.exe の登録 DdScm64.exe /a

DdScm64.exe の解除 DdScm64.exe /r

Windows Vista および Windows 7 での制限事項

Windows Vista および Windows 7 で DdScm.exe, DdScm32.exe, DdScm64.exe を実行すると確認ダイアログが表示されます。

● [プロテクト NT サービス]の登録

アプリケーションのインストールプログラム作成時に、インストール完了ダイアログが表示される前のタイミングなどで、自動的に登録されるようにインストールスクリプトに「DdScm.exe /a」、「DdScm32.exe /a」、「DdScm64.exe /a」のいずれかを加えてください。インストールスクリプトに DdScm.exe, DdScm32.exe, DdScm64.exe の登録を加えない場合は、インストール直後に (Administrators 権限下で) 手動で「DdScm.exe /a」、「DdScm32.exe /a」、「DdScm64.exe /a」のいずれかを実行する必要がありますが、ユーザに操作を指示することになるため、あまり現実的ではありません。また、開発時にアプリケーションテストを行う場合でも (Administrators 権限下で) 手動で「DdScm.exe /a」、「DdScm32.exe /a」、「DdScm64.exe /a」のいずれかを実行する必要があります。(ただし、ProtectKit3.5 をインストールした開発用 PC 上では通常、ProtectKit3.5 のインストーラによって自動的にこの [プロテクト NT サービス] が登録されています。)

この [プロテクト NT サービス] は、一度登録されれば、解除するまでサービスが継続されます。

Windows の再起動やシャットダウン後でも、Windows の起動時に自動的にサービスが開始されますので、その都度この [プロテクト NT サービス] を登録する必要はありません。

● [プロテクト NT サービス]の解除

作成したアプリケーションをアンインストールする際には「DdScm.exe /r」、「DdScm32.exe /r」、「DdScm64.exe /r」のいずれかを実行して、[プロテクト NT サービス] を解除してください。アンインストールプログラム開始直後 (アンインストーラによって「DdScm.exe」「DdScm32.exe」「DdScm64.exe」が削除される前) に解除するようにアンインストールスクリプトに加えてください。

アンインストールスクリプトに DdScm.exe, DdScm32.exe, DdScm64.exe の解除を加えない場合は、アンインストール直前に (Administrators 権限下で) 手動で「DdScm.exe /r」、「DdScm32.exe /r」、「DdScm64.exe /r」のいずれかを実行する必要がありますが、ユーザに操作を指示することになるため、あまり現実的ではありません。

■ アプリケーションを再作成した場合の配布方法

(1) ネイティブアプリケーション

アプリケーションに修正を施した場合、開発環境で作成した「プロテクト未設定アプリケーション(EXE)」に対し再度、プロテクト情報設定ツール(パスワード発行型)を使用してプロテクト情報を設定した「プロテクト完了アプリケーション(EXE)」を配布する必要があります。その場合、ユーザの環境に既に設定されているプロテクト情報を有効にし、再度プロテクトがかからないようにするため、プロテクト情報保持ファイルは配布しないようにしてください。

(2) CLR アプリケーション

運用上の注意事項

アプリケーションに修正を施した場合、そのアプリケーション(EXE)を配布する必要があります。その場合、ユーザの環境に既に設定されているプロテクト情報を有効にし、再度プロテクトがかからないようにするため、プロテクト情報保持ファイルは配布しないようにしてください。

■プロテクトアプリケーションの多重起動について

複数のインスタンスが実行された場合、一番最後に破棄されるインスタンスの設定が有効となります。たとえば、使用回数でプロテクトをかけ同時に 2 つのインスタンスを実行する場合、最後に終了するインスタンスが破棄される時に使用回数が 1 回分マイナスされます。つまり 2 回同時に実行しても使用回数は 1 回だけになります。

■プロテクトアプリケーション(CLR アプリケーション)の難読化の必要性

難読化されていないアプリケーションでは、逆コンパイルされた場合、コードや文字列が丸見えになってしまいます。それを防ぐため、アプリケーションの難読化を行うことをお勧めいたします。難読化を行う際には文字列およびリソースについても暗号化処理を行ってください。

2. ネットワーク型プロテクト

■ネットワークプロトコル

TCP/IP に対応しています。

■ネットワークの外部流出に注意

プロテクト情報設定ツール(ネットワーク型)で設定したネットワークは外部に流出しないように十分注意してください。

■プロテクトアプリケーションの多重起動について

基本的にアプリケーションの多重起動には対応しておりませんが、プログラミングで対応が可能です。詳しくは付属のサンプルをご覧ください。

■アプリケーションを再作成した場合の配布方法

アプリケーションに修正を施した場合、開発環境で作成した「プロテクト未設定アプリケーション(EXE)」に対し再度、プロテクト情報設定ツール(ネットワーク型)を使用してネットワークを設定した「プロテクト完了アプリケーション(EXE)」を配布する必要があります。

■製品固有情報の保存

自機 PC でライセンスを供与されているアプリケーションとネットワークに接続されている他 PC で稼働しているアプリケーションとを識別するために、製品のシリアル番号などの固有情報をレジストリに保存するようにしてください。ただし、固有情報を「ABC001,ABC002,・・・」のように設定すると、エンドユーザがこの情報を保存してあるレジストリキーを発見し、その内容を「ABC003」と変更することにより、プロテクトが解除されてしまいますので、規則性がわからないように設定してください。

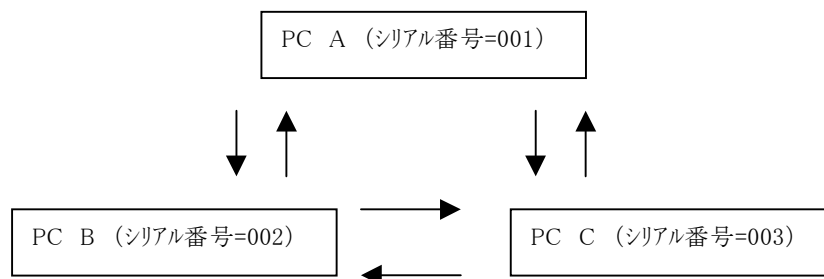
■送信メッセージ(PROTECTKIT3NET 構造体の szMessage)について

ネットワークに接続されている他 PC で稼働しているアプリケーションに送信するメッセージとして製品のシリアル番号などの固有情報を設定します。

■フォームのリサイズイベントで他 PC を監視

ネットワークに接続されている他 PC で稼働しているアプリケーションから受信されるメッセージ(製品のシリアル番号などの固有の情報)を取得し、自機 PC のシリアル番号などの固有の情報とチェックを行う必要があります。その際に他 PC で同じ固有情報を検出した場合、所定のレジストリに値を設定することにより LAN ケーブルをはずしても、アプリケーション起動時にレジストリの値をチェックすることにより、アプリケーションの起動を防ぐことができます。

メッセージの流れ



9. 作成したアプリケーションの配布

ProtectKit3/ProtectKitNET を使用して作成したアプリケーションをユーザに配布する場合は、そのアプリケーションの実行時に以下のファイルが必要になりますので、忘れずに配布リストに加えてください。

1. パスワード発行型プロテクトアプリケーション

A. ネイティブアプリケーション

(1) 32 ビットアプリケーション

ProFunc32.dll (ProtectKit3 パスワード発行型関数 DLL) またはリネームしたファイル

ProInfo32.dll (プロテクト情報保持ファイル) またはリネームしたファイル

ProInfo32.dll をリネームする場合はプロテクト情報設定ツール (パスワード発行型) で [名前を付けて保存] を選択して名称を変更してください。

DdScm32.exe または DdScm64.exe (アプリケーションをインストールする環境により必要なファイルは異なります) インストールする OS が不定 (32 ビット版なのか 64 ビット版なのか分からない) な場合は、DdScm.exe, DdScm32.exe, DdScm64.exe をすべて配布し、DdScm.exe を登録してください (DdScm.exe の内部で OS を判定し、対象となるプログラムを実行します)。

(2) 64 ビットアプリケーション

ProFunc64.dll (ProtectKit3 パスワード発行型関数 DLL) またはリネームしたファイル

ProInfo64.dll (プロテクト情報保持ファイル) またはリネームしたファイル

ProInfo64.dll をリネームする場合はプロテクト情報設定ツール (パスワード発行型) で [名前を付けて保存] を選択して名称を変更してください。

DdScm64.exe

B. CLR アプリケーション

(1) 32 ビットアプリケーション

Newtone.ProtectKit.Win.Password.dll (ProtectKitNET アセンブリ)

ProFuncNET32.dll (ProtectKitNET 用本体 DLL) またはリネームしたファイル

ProInfo32.dll (プロテクト情報保持ファイル) またはリネームしたファイル

ProInfo32.dll をリネームする場合はプロテクト情報設定ツール (パスワード発行型 - VS.NET 用) で [名前を付けて保存] を選択して名称を変更してください。

DdScm32.exe または DdScm64.exe (アプリケーションをインストールする環境により必要なファイルは異なります) インストールする OS が不定 (32 ビット版なのか 64 ビット版なのか分からない) な場合は、DdScm.exe, DdScm32.exe, DdScm64.exe をすべて配布し、DdScm.exe を登録してください (DdScm.exe の内部で OS を判定し、対象となるプログラムを実行します)。

(2) 64 ビットアプリケーション

Newtone.ProtectKit.Win.Password.dll (ProtectKitNET アセンブリ)

ProFuncNET64.dll (ProtectKitNET 用本体 DLL) またはリネームしたファイル

ProInfo64.dll (プロテクト情報保持ファイル) またはリネームしたファイル

ProInfo64.dll をリネームする場合はプロテクト情報設定ツール (パスワード発行型 - VS.NET 用) で [名前を付けて保存] を選択して名称を変更してください。

DdScm64.exe

上記のファイルは配布するアプリケーションと同じフォルダにコピーするようにインストーラに設定してください。

ただし、FD や DVD(CD)-ROM、ネットワークのリモートドライブなどでは動作いたしませんので、必ずローカルのハードディスクドライブにコピーしてください。(ドライブのルートは不可)

また、Windows のユーザ権限によっては書き込みのできないフォルダ (たとえば Windows のシステムフォルダ) は避けてください。

ProtectKitNET で提供しているアセンブリは厳密名付ですが、グローバルアセンブリキャッシュ (GAC) には登録しないでご利用ください。

CLR アプリケーションについては「12. CLR アプリケーションのビルドオプション」もご覧ください。

作成したアプリケーションの配布

2. ネットワーク型プロテクトアプリケーション

(1) 32 ビットアプリケーション

PkitNet32.dll (ProtectKit3 ネットワーク型関数 DLL) またはリネームしたファイル

(2) 64 ビットアプリケーション

PkitNet64.dll (ProtectKit3 ネットワーク型関数 DLL) またはリネームしたファイル

PkitNet32.dll/PkitNet64.dll ファイルは、配布するアプリケーションと同じフォルダもしくはパスの通ったフォルダ (Windows のシステムフォルダなど) にコピーするようにインストーラに設定してください。

再配布禁止ファイル(以下のファイルはその理由にかかわらず配布することを禁止しております。)

Pr3SetProtect.exe (ProtectKit3 プロテクト情報設定ツール[パスワード発行型] 32/64 ビット兼用)

Pr3SetProtectNET.exe (ProtectKit3 プロテクト情報設定ツール[ネットワーク型] 32/64 ビット兼用)

Pr3PassGen.exe (ProtectKit3 パスワード発行ツール 32/64 ビット兼用)

Pr3PassGen.dll (ProtectKit3 パスワード発行 DLL 32 ビット用)

Pr3PassGen.ocx (ProtectKit3 パスワード発行 OCX 32 ビット用)

PrNETSetProtect.exe (ProtectKitNET プロテクト情報設定ツール[パスワード発行型] 32/64 ビット兼用)

10. DdScm.exe,DdScm32.exe,DdScm64.exe のオプション機能について

DdScm.exe,DdScm32.exe,DdScm64.exe[プロテクト NT サービス]は、ProtectKit3 を利用したアプリケーションを実行する場合に必要なサービスです。

登録や解除、及び自動更新などの機能を/オプションにより実現します。

DdScm.exe を実行する場合は、同じフォルダに DdScm32.exe または DdScm64.exe が必要です。

1./i オプション (DdScm.exe /i, DdScm32.exe /i, DdScm64.exe /i)

DdScm32.exe,DdScm64.exe を登録します。既に登録されている場合は、登録回数に 1 足します。

このオプションでは、既に登録されているバージョンが有効となり、現在のバージョンで再登録を行いません。

2./r オプション (DdScm.exe /r, DdScm32.exe /r, DdScm64.exe /r)

DdScm32.exe,DdScm64.exe を解除します。既に登録されている場合は、登録回数から 1 引きます。

3./a オプション (DdScm.exe /a, DdScm32.exe /a, DdScm64.exe /a)

DdScm32.exe,DdScm64.exe を登録します。既に登録されている場合は、登録回数に 1 足します。

このオプションでは、既に登録されているバージョンを無効とし、現在のバージョンで再登録を行います。

この機能はターゲットとなる PC に既に ProtectKit3 を使用したアプリケーションが組み込まれていて、登録されている DdScm32.exe,DdScm64.exe のバージョンが最新でない場合にお勧めです。

4./u オプション (DdScm.exe /u, DdScm32.exe /u, DdScm64.exe /u)

既に登録されている DdScm32.exe,DdScm64.exe を更新します。

このオプションでは、登録回数には変更を加えず、既に登録されているバージョンを無効とし、現在のバージョンで再登録を行います。

この機能は DdScm32.exe,DdScm64.exe の不具合や機能拡張により更新された場合、ユーザに修正プログラムを配布する場合などにお勧めです。

5./m オプション(補助機能)

1~4 のオプションと組み合わせて使用します。このオプションは実行時に成功失敗などを表すメッセージボックスが表示されます。

11. ProtectKit2/ProtectKit2.5 との互換性

1. パスワード発行型プロテクトアプリケーション

(1)既に運用中のアプリケーション(ProtectKit2/2.5 で作成)を ProtectKit3 に移行する場合

(A)PC 識別条件に乱数情報を付加している場合(パスワードの有効期限ありの場合を含む)

ProtectKit3 に移行することができませんので、ProtectKit2/2.5 で運用されているアプリケーションをそのままご利用ください。

(B)(A)以外の場合

ProtectKit3 に移行することができます。

移行手順:

関数名や構造体名が変更されているため、アプリケーションを再構築します。

プロテクト情報設定ツールの「**パスワードの構成 - ProtectKit2.x のパスワードとの互換性**」を有効にします。

また、ProtectKit2/2.5 のアプリケーションとロックワード・PC 識別条件・パスワードの構成を同じにします。

ProtectKit3 のアプリケーションを配布し、パスワードを発行してプロテクトを解除します。

既にパスワードを発行していた場合(ProtectKit2/2.5 のアプリケーションでプロテクトを解除済み)は、以前発行したパスワードを使用してプロテクトを解除します。パスワードは ProtectKit2/2.5 付属のパスワード発行ツール(もしくは OCX や DLL)を使用してください。

(2)新たにアプリケーションを作成してインストールする場合

関数名や構造体名が変更されているため、アプリケーションを再構築する必要がありますが、関連するファイルを差し替えてお使いいただけます。

プロテクト情報設定ツールの「**パスワードの構成 - ProtectKit2.x のパスワードとの互換性**」を無効にすることをお勧めいたします。

2. ネットワーク型プロテクトアプリケーション

関数名や構造体名が変更されているため、アプリケーションを再構築する必要がありますが、関連するファイルを差し替えてお使いいただけます。

12. CLR アプリケーションのビルドオプション

Newtone.ProtectKit.Win.Password.dll について

プラットフォームターゲットを「any CPU」でビルドしてあります。

そのため、プロテクト対象アプリケーションのプラットフォームターゲットを「any CPU」でビルドした場合、32 ビット OS か 64 ビット OS かによって処理を分ける必要があります。

1.32 ビット OS の場合

DllFileName プロパティに ProFuncNET32.dll (名前を変える場合はその名前) を Initialize メソッドのパラメータ protectResDll に ProInfo32.dll (名前を変える場合はその名前) をそれぞれ設定する必要があります。

2.64 ビット OS の場合

DllFileName プロパティに ProFuncNET64.dll (名前を変える場合はその名前) を Initialize メソッドのパラメータ protectResDll に ProInfo64.dll (名前を変える場合はその名前) をそれぞれ設定する必要があります。

対処方法:

A. アプリケーションのコードでファイルを切り替える

稼働している OS が 32 ビットか 64 ビットかによって、DllFileName プロパティや Initialize メソッドのパラメータを変更する。

B. プロテクト対象アプリケーションのインストール時に該当するファイルをインストールする

一例ですが、32 ビットと 64 ビットの ProFuncNET32.dll/ProFuncNET64.dll および ProInfo32.dll/ProInfo64.dll をそれぞれ別フォルダに配置しておき (ファイル名は 32 ビットと 64 ビットで同じとする --> 同じコードで動作するため)、インストールする OS により該当するファイルをコピーする。

プロテクト対象アプリケーションのプラットフォームターゲットを「x86」または「x64」でビルドする場合は、それぞれに必要なファイルを設定するだけでよい (特殊な処理は不要)。

索引

.NET Framework コンポーネントを使用する場合 6

A

ActiveX コントロールの組み込み 5

D

DdScm.exe 5, 45, 47, 49
 DdScm.exe のオプション機能について 49
 DdScm32.exe 5, 45, 47, 49
 DdScm32.exe のオプション機能について 49
 DdScm64.exe 5, 45, 47, 49
 DdScm64.exe のオプション機能について 49

E

Eメールの送信 35, 44

L

lPasswordType (カスタムプロパティ) 41

N

Newtone.ProtectKit.Win.Password.dll 5, 6, 47, 51

P

PC 識別条件 20, 25
 PkitNet32.dll 5, 48
 PkitNet64.dll 5, 48
 Pr3PassGen.dll 33, 48
 Pr3PassGen.exe 9, 13, 32, 48
 Pr3PassGen.ocx 36, 48
 Pr3SetProtect.exe 8, 19, 20, 48
 Pr3SetProtectNET.exe 14, 29, 48
 PrNETSetProtect.exe 11, 24, 48
 ProFunc32.dll 5, 47
 ProFunc64.dll 5, 47
 ProFuncNET32.dll 5, 47
 ProFuncNET64.dll 5, 47
 ProInfo32.dll 5, 19, 20, 24, 47
 ProInfo64.dll 5, 19, 20, 24, 47
 ProtectKit2/ProtectKit2.5 との互換性 50
 ProtectKit35.exe 5
 PS3GetPassword 34
 PS3GetPassword (カスタムメソッド) 43
 PS3SendMail 35
 PS3SendMail (カスタムメソッド) 44

S

szFromAddr (カスタムプロパティ) 37

szFromHdr (カスタムプロパティ) 37
 szLockword (カスタムプロパティ) 38
 szMailSvrAddr (カスタムプロパティ) 37
 szMessage (カスタムプロパティ) 39
 szPassword (カスタムプロパティ) 40
 szPcID (カスタムプロパティ) 42
 szSubject (カスタムプロパティ) 37
 szToAddr (カスタムプロパティ) 37
 szToHdr (カスタムプロパティ) 37

W

Windows 7 での制限事項 45
 Windows Vista での制限事項 45

あ

アプリケーションの配布 47

い

インストール後のフォルダの構成 5
 インストール方法 5

う

運用上の注意事項 45

さ

再配布禁止ファイル 48
 サンプルプログラム 7

し

実行監視イベント 14
 使用回数 21, 26
 使用期限 21, 26
 使用月数 21, 26
 使用日数 21, 26

た

対応開発コンテナ 3
 対応プラットフォーム 3
 多重起動 46

と

問い合わせ先 4

な

難読化の必要性 46

ね

ネットワークプロトコル	14, 46
ネットワーク型プロテクト.....	46
ネットワーク型プロテクト(ネイティブアプリケーション)	14
ネットワークワード	15, 30, 46

は

パスワードの期限あり.....	22, 26
パスワードの構成	22, 26
パスワードの互換性	22, 26, 31, 50
パスワードの発行	32, 34, 43
パスワード発行ツール.....	9, 13
パスワード発行について.....	31
パスワード発行 OCX を使用する場合	5
パスワード発行型プロテクト.....	45
パスワード発行型プロテクト(CLR アプリケーション) 10	
パスワード発行型プロテクト(ネイティブアプリケーション)	8

ひ

ビルドオプション	51
----------------	----

ふ

ファイル	22, 26
プロテクトアプリケーションの運用手順	8
プロテクト完了アプリケーション	19, 29, 45, 46
プロテクト情報設定ツール	19
プロテクト情報設定ツール(ネットワーク型)	14, 29
プロテクト情報設定ツール(パスワード発行型)8, 11, 19, 24	
プロテクト情報保持ファイル	19, 20, 24
プロテクト方法.....	9, 12, 21, 26
プロテクト未設定アプリケーション	19, 29, 45, 46

ら

乱数情報	20, 25
------------	--------

れ

レジストリ.....	22, 26
レジストリへの登録及び解除	6

ろ

ロックワード	9, 12, 21, 25, 32, 45
--------------	-----------------------

ProtectKit3.5
プログラミングマニュアル

©2010 NEWTONE Corp.

発行 株式会社ニュートン

〒940-0015
新潟県長岡市寿 1-6-43
TEL 0258-24-7900
FAX 0258-24-7905
<http://www.newtone.co.jp/>