HOW TO IMPLEMENT CUSTOM LICENSING FOR ACTIVEX LIBRARIES

Editor:

Tomáš Mandys, tomas.mandys@2p.cz (2p plus)

Home site:

http://www.2p.cz

Document status:

Version 1.0 First release

Table of Contents

Disclaimer

The information of this document is provided ‚“AS IS‘, with no warranties whatsoever, excluding in particular any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. This document is provided for information purposes only.

1. Introduction

An ActiveX library support simple licensing procedure using a license file (*.LIC*). The license file contains a license string. When the ActiveX is loaded into memory and license is required library typically looks for file that has the same name as library but with extension *.LIC* and compares license string to a build in license key. If license is bad that library is not loaded.

But this behavior does not enable online licensing of developers libraries. Such library can be used by developer for free but cannot be deployed without valid license.

2. Custom registration

Typical example:

1. Developer downloads an ActiveX developer library
2. Developer install library to IDE (Visual Basic, Delphi, Microsoft Excel, etc.)
3. Library displays registration dialog to register and generates machine unique key that is sent to library provider
4. The library provider will send activation key to developer
5. Developer can use ActiveX library but cannot deploy application based on the library
6. Developer buy deploy key from library provider and deploy own applications

2.1. AxCtrlsReg.pas

The *AxCtrlsReg* defines new *TRegActiveXControlFactory* class that implements registration functionality.

```
unit AxCtrlsReg;

interface
uses
  Classes, Windows, SysUtils, ComObj, Controls, AxCtrls;

type
  TRegActiveXControlFactory = class(TActiveXControlFactory)
  protected
    function ValidateUserLicense(const LicStr: WideString): Boolean; override;
    function HasMachineLicense: Boolean; override;
  public
    class function CreateLicString(AppKey: Word; const AppName, SecretKey: string;
AppId: Word; Expiration: Word; ...): string;
  end;

implementation
uses
  <CustomRegistrationTool>;


{ TActiveXControlRegFactory }


class function TRegActiveXControlFactory.CreateLicString;
begin
  Result:=
{ create registration string using custom algorithm – combine all parameters into one
string }
end;

function TRegActiveXControlFactory.ValidateUserLicense(const LicStr: WideString):
Boolean;
begin
{ if LicStr or GetLicenseFileName file is a valid non expired license then return
True otherwise return False, do not allow a user interaction, dialog, etc. }
end;

function TRegActiveXControlFactory.HasMachineLicense: Boolean;
var
  L: Word;
begin
  Result:= not SupportsLicensing;
  if Result then
    Exit;
{ display a registration dialog
  if customer passes registration successfully
  write registration string to GetLicenseFileName file and
  return True }
end;

end.
```

2.2. *Custom component implementation*

Implementation is generated by a wizard or written by developer. Just add *AxCtrlsReg* and change ActiveX factory registration.

```
unit MyCustomImpl;

interface

uses
  Windows, ComObj, ActiveX, AxCtrls, AxCtrlsReg, Classes, StdVcl, MyCustom_TLB;

type
  TMyCustomX = class(TActiveXControl, IMyCustomX)
  private
    FEvents: IMyCustomXEvents;
  ...
  end;

implementation

...

initialization
  TRegActiveXControlFactory.Create(
    ComServer,
    TMyCustomX,
    Class_MyCustomX,
    1,
    TActiveXControlFactory.CreateLicString(...),
    0,
    tmBoth);
end.
```

2.3. *Type library*

Note that Delphi ActiveX import wizard (*Component/Install ActiveX Control*) imports license key into init control data record.

```
procedure TGSMX.InitControlData;
const
  CEventDispIDs: array [0..5] of DWORD = (
    $00000001, $00000002, $00000003, $00000004, $00000005, $00000006);
  CLicenseKey: array[0..310] of Word = (
    , $0063, $0065, $002E, $0068, $0074, $006D, $006C, $003F, $0052, $0045
    , $0074, $0070, $003A, $002F, $002F, $0077, $0077, $0077, $002E, $0032
    ...
  );

  CControlData: TControlData2 = (
    ClassID: '{1B7EF3BB-6790-42B0-8B0E-5044645577FA}';
    EventIID: '{7325ED3A-FE23-45A8-9885-0F0444877478}';
    EventCount: 6;
    EventDispIDs: @CEventDispIDs;
    LicenseKey: @CLicenseKey;
    Flags: $00000000;
    Version: 401);
begin
  ControlData := @CControlData;
  TControlData2(CControlData).FirstEventOfs := Cardinal(@@FOnRxChar) -
Cardinal(Self);
end;
```