

ZQure Version 1.0

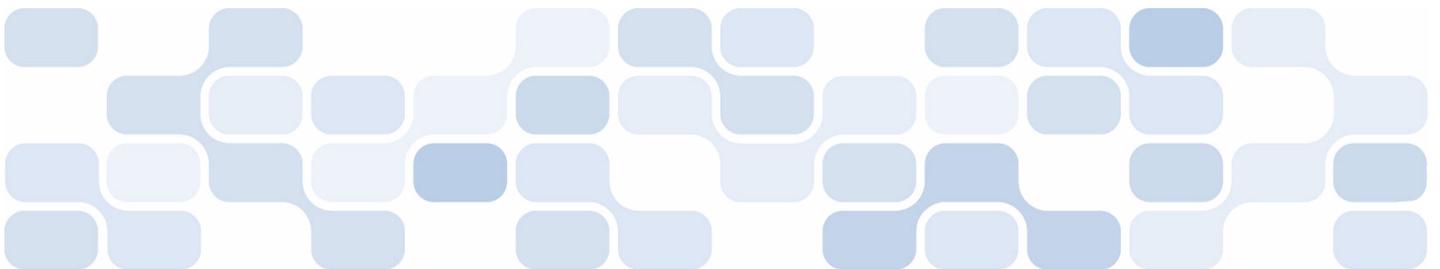
Software Licensing and Protection

October 2009

Thayne Breetzke

Copyright © 2009 Sophic Edge Solutions.

All rights reserved.



This documentation is for informational purposes only. The author MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of the author.

Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events that may be depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Copyright © 2009 Sophic Edge Solutions. All rights reserved.

Microsoft, Windows, Windows Server, Windows Vista, Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.

Product Support and Feedback

E-mail: thayne@SophicEdge.com

Web: www.SophicEdge.com

Table of Contents

1 Introduction.....	5
1.1 System Requirements	6
1.2 What Does ZQure Comprise Of?	6
1.3 What is a Licence Key?	8
2 Quick Start Tutorial	10
2.1 Add a ZQure Application	10
2.2 Decide on the Form of Licence Key	11
2.3 Create the Visual Studio Application	12
2.4 Configure the Main Form	13
2.5 Create the <i>Registration</i> Form	13
2.6 Create the <i>Enter Licence Key</i> Form	16
2.7 Save the Licence Key upon Registration.....	18
2.8 Retrieve the Licence Key on Startup.....	19
2.9 Test the Application	21
3 The ZQure Management Tool	28
3.1 Managing Applications.....	30
3.2 Managing Licence Keys.....	32
3.3 Using the Tools.....	35
3.4 Viewing Help	36
3.5 Registering ZQure.....	37
4 The ZQure Registration Library.....	40
4.1 <i>RegKey</i> Constructor	42
4.2 <i>CreateConfigDoc</i> Method	43
4.3 <i>GetComputerID</i> Method	44
4.4 <i>ReadConfigFile</i> Method	45
4.5 <i>RegInfoFromXML</i> Method	46
4.6 <i>RegInfoToXML</i> Method.....	47
4.7 <i>ToString</i> Method	48
4.8 <i>Validate</i> Method.....	49
4.9 <i>WriteConfigFile</i> Method.....	50
4.10 <i>Value</i> Property.....	52
4.11 <i>defaultConfigAppComment</i> Constant	53

4.12 <i>defaultConfigLibComment</i> Constant.....	53
4.13 <i>KeyStatus</i> Enumerated Type	54
4.14 <i>keyStruct</i> Structure.....	54
4.15 <i>userStruct</i> Structure.....	55
4.16 <i>configKey</i> Class	58
5 Tips for Developers.....	60
6 ZQure Licences	61
6.1 ZQure Evaluation Licence.....	61
6.2 ZQure Standard Licence	63

1 Introduction

Software piracy is the illegal use, reproduction and/or distribution of copyrighted software applications. Many users of software applications do not realize that the applications they purchase do not belong to them. Instead, each purchaser becomes a licensed user of the application. Unless stated otherwise in a software package's licence agreement, the purchaser purchases the right to use the software on a single computer. Making copies of the software and placing them on other computers or passing the software on to others is software piracy. Whether deliberate or not, software piracy is illegal and punishable by law.

Obviously, software piracy hurts software developers as it has an enormous negative impact on them financially. It is believed that software piracy has led to a worldwide economic loss estimated to be in the tens of billions of US dollars. Devising ways to protect software can be frustrating and can unnecessarily consume developers' valuable time and money.

ZQure can assist in combating this problem. It is a software licensing and protection tool for .NET developers that helps software developers easily protect their desktop applications from being distributed and used unlawfully. Essentially, each purchaser of the developer's software is issued with a licence key generated by ZQure. The licence key is used to activate the user's copy of the software. In this way, a developer might distribute a "crippled", but free, evaluation version of a software product for evaluation by potential users. Users can then opt to purchase a licence key to immediately activate the full functionality of the product. Alternatively, the product can be made completely unusable until it is activated with a licence key. Licence keys can optionally be made to expire after a period of time creating the possibility of enforcing the re-licensing of a product on a recurring basis.

ZQure offers a host of benefits, including the following:

- It helps prevent illegal distribution of your software by preventing execution of your software if it has been copied from another computer.
- It features exceptionally strong multi-layer protection to ensure that only legally licensed users can access the protected software applications based on electronic security keys.
- It offers quick and easy licence key generation.
- It offers an easy-to-use licence key manager.
- It offers simple and straightforward software-based product activation.
- It is entirely software based, requiring no hardware dongles, special disks or other external tools.
- It provides software developers full control of their licence creation and distribution.
- It offers highly flexible software licence restrictions. You can:
 - Optionally restrict key usage on a per computer basis.
 - Optionally restrict key usage on a per software product basis.
 - Optionally impose a licence key expiry date.
- It includes tamper detection to respond to software piracy.

This documentation provides thorough coverage of how ZQure is used to enforce software licensing and protection of your software.

1.1 System Requirements

The following are the system requirements for ZQure:

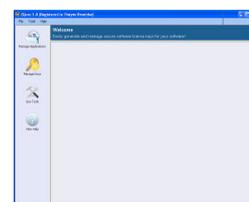
Operating System	Windows XP with SP2, Windows Server 2003 or 2008, or Windows Vista (for a specific list of supported operating systems, please refer to the requirements for the .NET Framework version 3.5)
.NET Framework	Version 3.5
IDE	Microsoft Visual Studio .NET 2005 or 2008
Processor	At least 400MHz, preferably 1GHz
RAM	At least 96MB, preferably 256MB
Hard Disk Space	5MB (excluding space for the .NET Framework version 3.5)
Software	Adobe Acrobat Reader (for reading the documentation)

Please note that the ZQure installation package does not include the .NET Framework version 3.5. When installing ZQure, you can optionally download and install the Framework if it is not already present on the destination computer.

1.2 What Does ZQure Comprise Of?

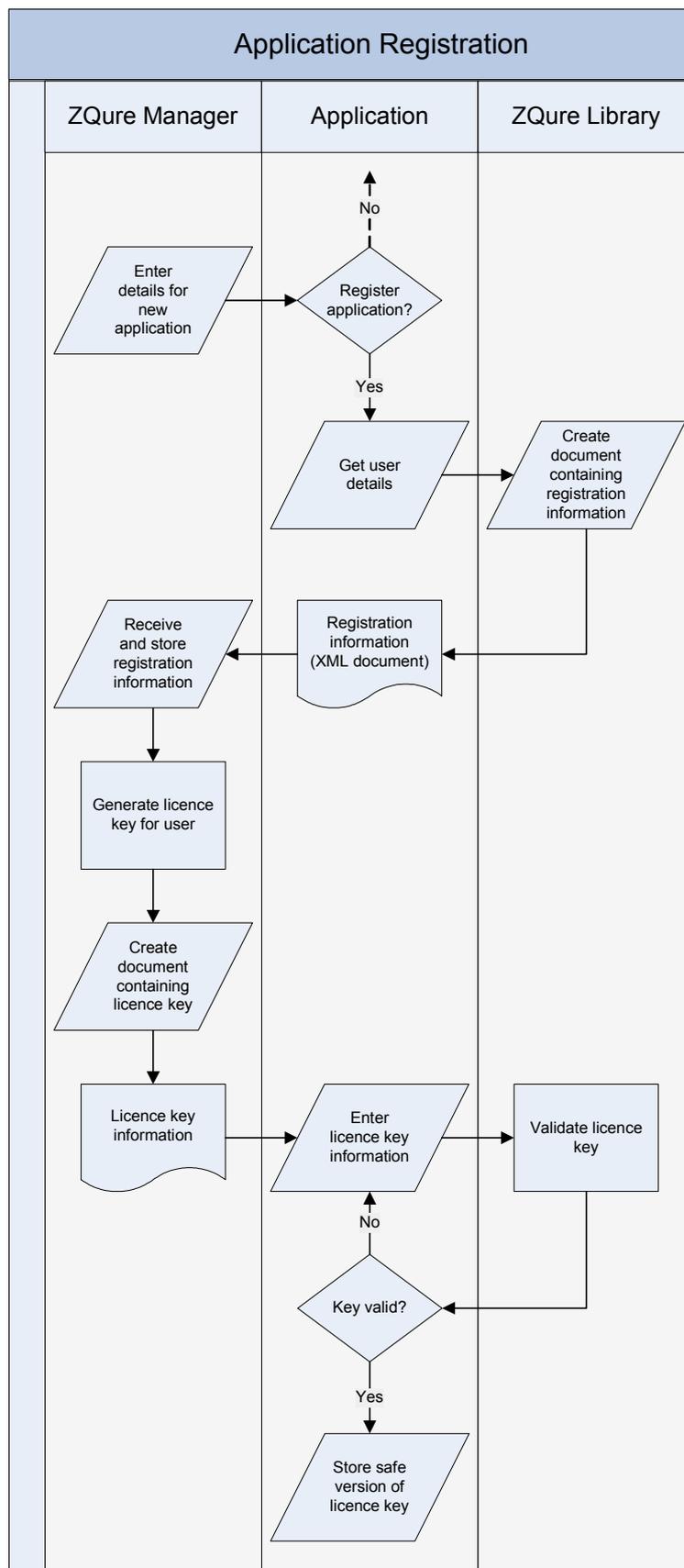
ZQure is comprised of two parts:

- **The ZQure Management Tool.** The ZQure management tool (or manager) allows the software developer to manage a list of applications for which licence keys can be generated and to create and manage the licence keys issued for each application. It also records the details of each user to whom a key has been issued.
- **The ZQure Registration Library.** The registration library is a .DLL (or component) that is used by the software developer's application to validate licence keys and to perform various other useful functions.

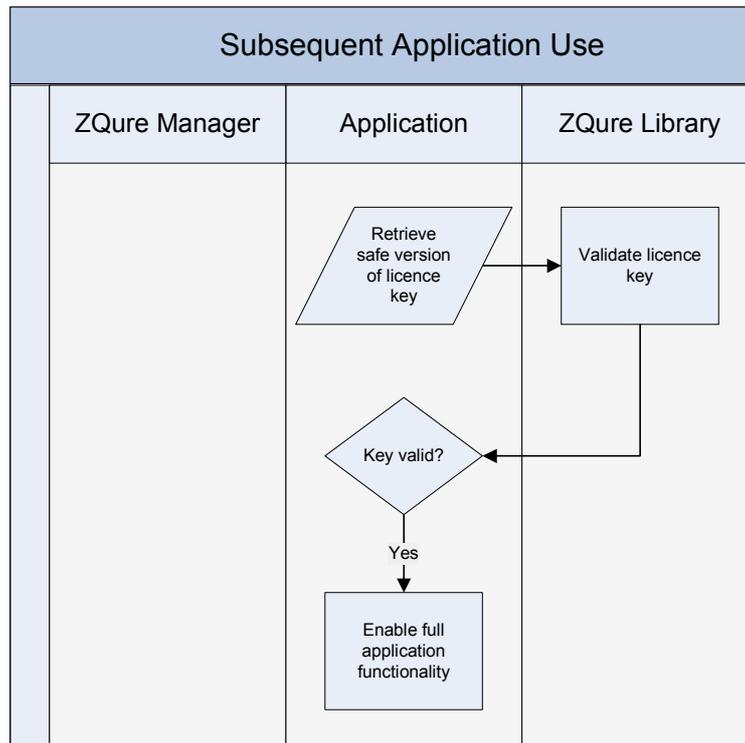


The two flowcharts that follow show the relationship between these two components in terms of the following two primary tasks:

- **Registering an application.** Before registration with a valid licence key, an application might be completely unusable or it might be an evaluation version that users can evaluate before purchasing the software. After successful registration, the application's full functionality can immediately be made available without the need for installing another version of the software.
- **Using the application after registration.** As long as the licence key is available to the application (e.g. by storing a safe version of it in a database or configuration file), the application can validate it each time the application is used. The application can therefore remain fully functional as long as the key is present and valid.



Flowchart: Registering an Application



Flowchart: Using an Application after Registration

PLEASE NOTE

Please read the ZQure licence agreements at the end of this document. By using this software, you agree to abide by the relevant licence agreement.

1.3 What is a Licence Key?

The outward appearance of a typical licence key looks as follows:

7AA9B9-CC4C76-6F8250-52173A-5622CB-D9CE24-83F868-909E90-0C95E9-801402-72F9F4-F43C8C-F46C3F-031DB0-18FA

Deciding on the suitable form of a licence key and the restrictions it will impose is a very important step requiring careful thought. A licence key is created based upon the following four elements:

- **Username (required).** The name of the user who the key has been issued to (i.e. the key holder).
- **Computer ID (optional).** An optional identifier that ties the licence key to a specific physical entity, such as a computer or network. Often, a licence key must be valid only on a specific computer (e.g. the computer on which an application was registered). In this case, an identifier unique to that computer is included. This prevents the

application from being copied and used on another computer. Sometimes, a licence key must be valid on any computer on a specific network. In this case, an identifier unique to that network, but the same across all its computers, is included. This computer ID element is versatile as it is not limited to a computer or network identifier. Any identifier that you wish to tie the licence key to may be used. For example, a network login name could even be used.

- **Application ID (optional).** An optional identifier that ties the licence key to a specific application. When specified, this is the *Application ID* provided for an application when specifying the details of a new application using the ZQure manager. An application ID is usually specified for a licence key, although it is not required. If not specified, the licence key can be used for any application that is protected with ZQure.
- **Expiry date (optional).** An optional expiry date for the licence key. Once a licence key has expired, it becomes invalid and unusable. For example, a user might purchase an evaluation licence key that is valid for a certain period of time. The application that requires the key will be usable until the key reaches its expiry date.

Together, the above elements define the scope of the licence key, which can be extremely wide (e.g. if only a username is specified) or very specific (e.g. if all four elements are specified). Generally, the scope of a licence key is made as specific as possible by specifying a username, computer ID and application ID, and specifying an expiry date if it is applicable.

The licence key, once generated, is issued to the user so that he/she can register his copy of the relevant application. It is kept safe and secret at all times. Because the key cannot be stored in its original form within the application where it might be found and used by authorized users, it is first transformed into a safe version by using a method in the ZQure library. A user cannot register with a safe version of a key.

The following section presents a quick start tutorial to demonstrate how easy it is to use the ZQure management tool and registration library to protect your software. A more detailed explanation of each of these two components and how they should be used is then given.

2 Quick Start Tutorial

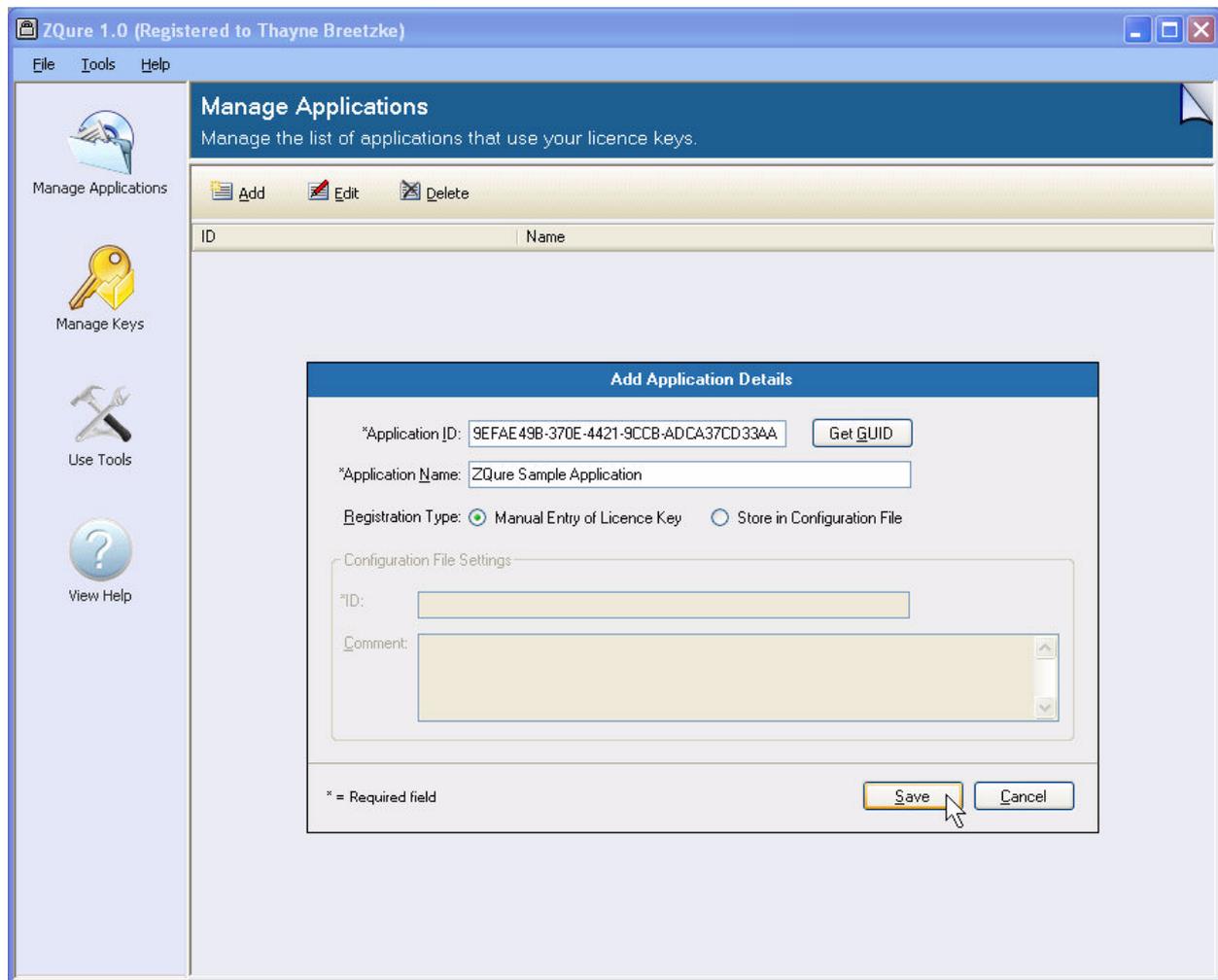
The following tutorial takes you through the simple steps that were followed to create the sample application provided in the **Sample Application** folder. The following is an overview of these steps:

1. Add a ZQure application
2. Decide on the form of licence key
3. Create the Visual Studio application
4. Configure the main form
5. Create the *Registration* form
6. Create the *Enter Licence Key* form
7. Save the licence key upon registration
8. Retrieve the licence key on startup
9. Test the application

2.1 Add a ZQure Application

The first step is to use the ZQure manager to add a record for the sample application. Follow these steps:

1. Start the ZQure manager.
2. Click *Manage Applications* and then the *Add* button. An application ID will be automatically generated for the new application. You can accept the default one or enter your own (any string value). A new one can be generated by clicking the *Get GUID* button. For this application, use the default one.
3. Enter "ZQure Sample Application" for the application's name.
4. Accept the default registration type where the user will enter a licence key manually to register the software. In contrast, the configuration file option allows the user to register the software by saving a generated configuration file to the software's application folder. The file would contain a safe version of the licence key generated for the user.
5. Click the *Save* button.



2.2 Decide on the Form of Licence Key

This sample application uses a non-expiring licence key based upon username, computer ID and application ID:

- The username is supplied by the user (e.g. Joe Soap).
- The computer ID is the unique identifier of the computer on which the application will be registered. Thus, the application, when registered, would only be usable on the computer on which it was registered. This identifier is generated using the `GetComputerID` method (an example of a computer ID is 2DA64EB8A64C3A7F463C53F59F549735688F236A).
- The application ID is the identifier that was assigned to the application using the ZQure manager (e.g. 9EFAE49B-370E-4421-9CCB-ADCA37CD33AA).

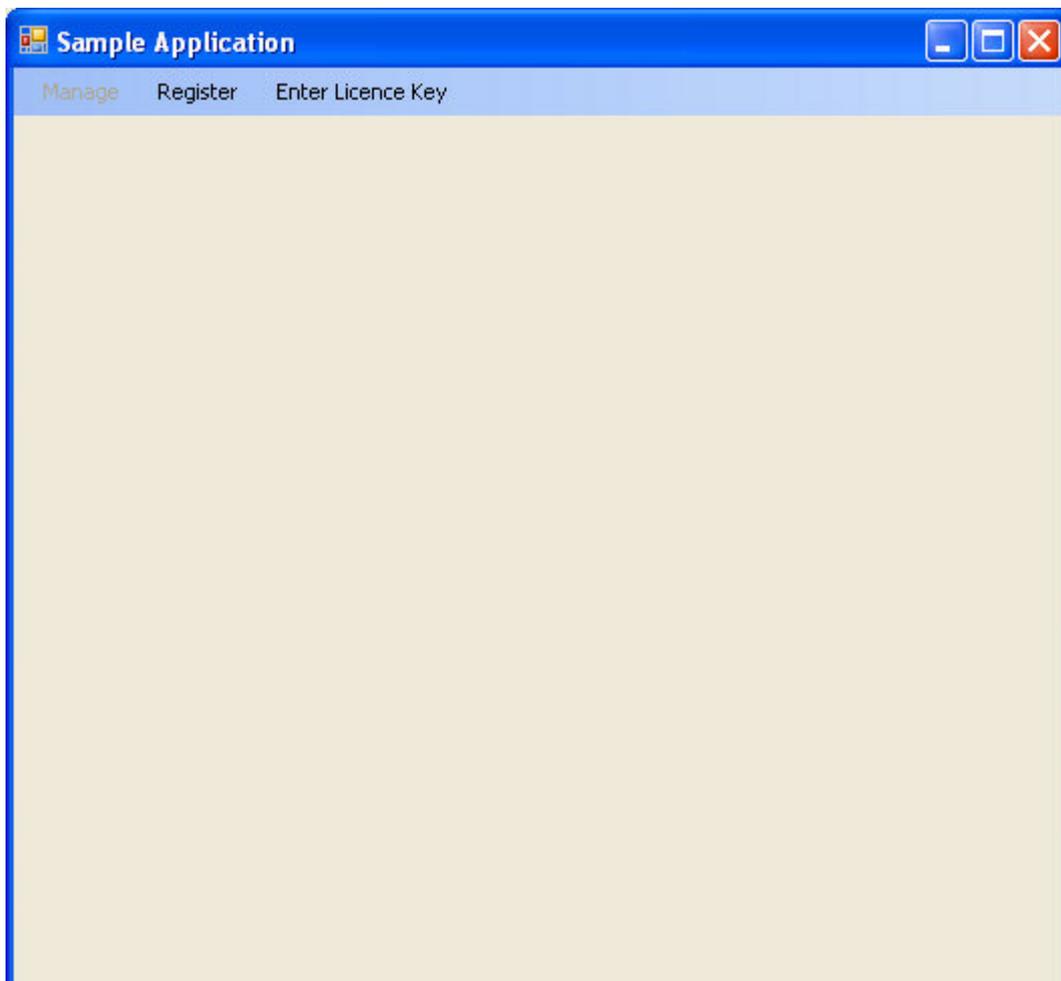
The chosen characteristics of the licence key are applied in the subsequent steps.

2.3 Create the Visual Studio Application

Follow these steps to create a new Visual Studio application that uses the ZQure Registration library:

1. Create a new Visual Studio application.
2. Add a reference to the **Registration.dll** component in the **Library** folder.
3. If you have registered ZQure, there will be a **Registration.dll.config** file in the **Library** folder. Add it to the solution (right click *Sample Application* | *Add | Existing Item...*) and set its *Copy to Output Directory* property to *Copy Always*. If you do not add the configuration file, or if you are using the evaluation version of ZQure, the **Registration** component will not be able to validate licence keys. Be sure that the **Registration.dll** and **Registration.dll.config** files are copied to the output folder after building the project.

The following step creates the main form for the application. It will contain a menu and look like this:



2.4 Configure the Main Form

Follow these steps to configure the application's main form:

1. Rename **Form1.cs** to **frmMain.cs** and change its *Text* property to "Sample Application".
2. Add a MenuStrip control to the form and name it **msMain**.
3. Add the following two menu items to the menu:
 - a. Manage (name it **mnuManage** and set its *Enabled* property to False).
 - b. Register (name it **mnuRegister**).
 - c. Enter licence key... (name it **mnuEnterKey**).

2.5 Create the *Registration* Form

Follow these steps to create the form that the user will use to register his/her copy of the application:

1. Add a new Windows form to the application and name it **frmRegister**.
2. Change the form's *Text* property to "Register".
3. Add the necessary controls to the form so that it looks as shown below.
4. Add the following code to **frmMain** to open **frmRegister** when the *Register* menu item is clicked:

```
private void mnuRegister_Click(object sender, EventArgs e)
{
    frmRegister frm = new frmRegister();
    frm.Show();
}
```

Important: Your e-mail address will be used to send you your licence key.

*Name:

*E-mail Address:

*Phone No: Fax No:

*Address:

*City: *Postal Code/Zip:

*Country:

Company:

* = Required field

Please e-mail the following text to the developer in order to receive your licence key:

5. Add the following code to **frmRegister** to handle a click of the *Register* button:

```
using Registration;

private void btnRegister_Click(object sender, EventArgs e)
{
    txtRegInfo.Text = RegKey.RegInfoToXML(new RegKey.userStruct (
        "ZQure Sample Application", txtName.Text,
        RegKey.GetComputerID(), "9EFAE49B-370E-4421-9CCB-
        ADCA37CD33AA", txtEmailAddr.Text, txtPhoneNo.Text,
        txtFaxNo.Text, txtAddrLine1.Text, txtAddrLine2.Text,
        txtCity.Text, txtPostalCodeZip.Text, txtCountry.Text,
        txtCompany.Text));
}
```

The following shows an example of the output that this statement produces:

Important: Your e-mail address will be used to send you your licence key.

*Name:

*E-mail Address:

*Phone No: Fax No:

*Address:

*City: *Postal Code/Zip:

*Country:

Company:

* = Required field

Please e-mail the following text to the developer in order to receive your licence key:

```
<registration>
<application>ZQure Sample Application</application>
<username>Joe Soap</username>
<computerID>2DA64EB8A64C3A7F463C53F59F549735688F236A</computerID>
<applicationID>D7963AA13A9DFAA6394DEE53D955F254E956E94D</applicationID>
<email>joe@mydomain.com</email>
<phone>[123] 456 7890</phone>
<fax></fax>
<address1>1 2nd Avenue</address1>
<address2>Thirstyville</address2>
<city>Port Fifth</city>
<code>1234</code>
<country>South Africa</country>
<company></company>
</registration>
```

[RegInfoToXML](#) receives user and key details and generates the XML to make sending this information to the developer easy. Take note of the following:

- The first argument in the `userStruct` constructor is a description of the application used for informational purposes only.
- Only include the computer ID argument in the `userStruct` constructor if it will be used to form the licence key. If it is not used, pass an empty string. In this example, a computer ID will form part of the licence key. `GetComputerID` is used to obtain a computer ID that is unique to the computer.

- Only include the application ID argument in the `userStruct` constructor if it will be used to form the licence key. If it is not used, pass an empty string. In this example, an application ID will form part of the licence key. The application ID argument is the identifier for the application, as shown in the ZQure manager. Notice that the application ID is automatically encrypted in the XML output for added security.

In this example, the user will copy the generated XML text to the clipboard and e-mail it to the developer in order to obtain a licence key.

2.6 Create the *Enter Licence Key* Form

The developer uses the ZQure manager to generate a licence key from the XML data received from the user. Once the user receives the key, he/she needs to enter the key using the *Enter Licence Key* form. Follow these steps to create this form:

1. Add a new Windows form to the application and name it **frmEnterKey**.
2. Change the form's *Text* property to "Enter Licence Key".
3. Add the necessary controls to the form so that it looks as follows:



4. Add the following code to **frmMain** to open **frmEnterKey** when the *Enter Licence Key* menu item is clicked:

```
private void enterLicenceKeyToolStripMenuItem_Click(object sender,
EventArgs e)
{
    frmEnterKey frm = new frmEnterKey();
    frm.Show();
}
```

5. Add the following code to **frmEnterKey** to handle a click of the *OK* button:

```
using Registration;

private void btnOK_Click(object sender, EventArgs e)
{
    //Create a licence key instance for the key string entered
    RegKey key = new RegKey(txtLicenceKey.Text.Trim());
}
```

```

//Validate the key
switch (key.Validate(txtRegisteredName.Text.Trim(),
    RegKey.GetComputerID(),
    "9EFAE49B-370E-4421-9CCB-ADCA37CD33AA", true))
{
    case RegKey.KeyStatus.Valid:
        ((MenuStrip)Owner.Controls["msMain"]).
            Items["mnuManage"].Enabled = true;

        ((MenuStrip)Owner.Controls["msMain"]).
            Items["mnuRegister"].Enabled = false;

        ((MenuStrip)Owner.Controls["msMain"]).
            Items["mnuEnterKey"].Enabled = false;

        this.Close();
        break;
    case RegKey.KeyStatus.Expired:
        MessageBox.Show("The licence key that you entered has
            expired.", "Expired Licence Key",
            MessageBoxButtons.OK);
        break;
    case RegKey.KeyStatus.ConfigError:
        MessageBox.Show("A configuration file error occurred.
            Please ensure that the required configuration file is
            present and valid.", "Error",
            MessageBoxButtons.OK);
        break;
    case RegKey.KeyStatus.Error:
        MessageBox.Show("An unexpected error occurred. Please
            try again. If the problem persists, please inform the
            developer.", "Error", MessageBoxButtons.OK);
        break;
    default:
        MessageBox.Show("The licence key is invalid. Please
            check that the name and licence key that you entered
            EXACTLY matches the name and licence key given in the
            registration document that was sent to you.", "Invalid
            Licence Key", MessageBoxButtons.OK);
        break;
}
}

```

After instantiating the `RegKey` class with key string argument in the constructor, you can check whether the key is a valid key by using the `Validate` method, as shown in the above code. The method expects the following arguments: username, computer ID and application ID, in that order. The fourth argument specifies whether the application is registering the user with the licence key.

As this sample application uses a non-expiring licence key based upon username, computer ID and application ID, username, computer ID and application ID arguments are provided to the `Validate` method. Depending on the form of licence key chosen for your application (see the second step), you might provide an empty string as one or more of the arguments. For example, if you had chosen not to use a computer ID for your application's licence key, an empty string would be provided as the computer ID argument. In this example, because the user is registering with the licence key, the fourth argument is `True`.

`Validate` returns a value indicating the success or failure of the validation check. In this case, if the validation succeeds, registration is deemed successful and the *Manage* menu on the main form is enabled, thus allowing the user full access to the application.

2.7 Save the Licence Key upon Registration

In the previous step, the *Manage* menu item was enabled if the licence key was valid for registering with. Obviously, the user cannot be expected to enter his/her licence key each time the application is run in order to access the *Manage* menu item. Therefore, the licence key needs to be recorded and re-validated each time the application executes. The key is usually stored in a database or a configuration file.

As a licence key must be kept safe and secret at all times, the key cannot simply be stored in its original form and where it might be found and used by authorized users. Therefore, whenever storing a key value for future use, use the `ToString()` method of a key instance to return a safe version of the key. A user cannot register with a safe version of a key.

In this sample application, the licence key is saved in the application's configuration file. Follow these steps to save the licence key:

1. Write the following code to create the configuration file:

```
private bool CreateConfigFile(string username, RegKey key)
{
    bool successful = true;

    List<RegKey.configKey> configKeys = new List<RegKey.configKey>();
    RegKey.configKey configKey = new RegKey.configKey();
    configKey.ID = "SampleAppKeyID";
    configKey.Type = "Application";
    configKey.KeyHolder = username;
    configKey.Value = key.ToString();
    configKey.Data = "";
    configKey.Comment = RegKey.defaultConfigAppComment.
        Replace("&keyHolder;", username);
    configKeys.Add(configKey);
    try
    {
        RegKey.WriteConfigFile(Application.ExecutablePath +
            ".config", null, configKeys);
    }
    catch (Exception ex)
    {
        successful = false;
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK);
    }
    return successful;
}
```

This code creates an instance of a configuration file key and adds it to a generic list. The `WriteConfigFile()` method writes the keys in the generic list to the specified configuration file. The following is an example of the resultant configuration file:

```

<configuration>
  <licenceKeys>
    <!--Key holder: Joe Soap
    If you, or the person in charge of this software, are not the key holder named
    above, this software is being used ILLEGALLY. Unauthorized use, reproduction
    or distribution of this key or this software, or any portion of it, will be
    prosecuted to the maximum extent possible under the law.-->
    <key id="SampleAppKeyID" type="Application" keyHolder="Joe Soap"
      value="8A0D60-6AFBED-4319EE-92C014-1F20E0-9B6128-94F405-26AF85-
      BDED52-52CA84-5AC31C-7D8438-76AD85-451DB0-18FC" data="" />
  </licenceKeys>
  <!--Other configuration settings-->
</configuration>

```

For further details of the methods used in this code, please refer to the next section entitled *The ZQure Registration Library*.

2. Call the `CreateConfigFile()` function in the `switch` statement and display the user's name in the main form's title bar if the function call is successful:

```

case RegKey.KeyStatus.Valid:
  if (CreateConfigFile(txtRegisteredName.Text.Trim(), key))
  {
    Owner.Text += " (Registered to " + txtRegisteredName.Text.Trim()
      + ")";

    ((MenuStrip)Owner.Controls["msMain"]).
      Items["mnuManage"].Enabled = true;

    ((MenuStrip)Owner.Controls["msMain"]).
      Items["mnuRegister"].Enabled = false;

    ((MenuStrip)Owner.Controls["msMain"]).
      Items["mnuEnterKey"].Enabled = false;

    this.Close();
  }
  break;

```

2.8 Retrieve the Licence Key on Startup

Each time the application starts, the licence key must be read from the configuration file. If the key is valid, the *Manage* menu item must be enabled. This is carried out in **frmMain**'s `Load()` event handler, as shown in the steps below:

1. Write the following code in the `Load()` event handler:

```

using Registration;
using System.Xml;

```

```
private void frmMain_Load(object sender, EventArgs e)
{
    try
    {
        XmlDocument readDoc = new XmlDocument();
        List<RegKey.configKey> configKeys = new
            List<RegKey.configKey>();
        RegKey.ReadConfigFile(Application.ExecutablePath + ".config",
            out readDoc, out configKeys);

        RegKey.configKey configKey = configKeys.Find(IDEquals);

        if (configKey != null)
        {
            RegKey key = new RegKey(configKey.Value);
            if (key.Validate(configKey.KeyHolder,
                RegKey.GetComputerID(),
                "9EFAE49B-370E-4421-9CCB-ADCA37CD33AA", false) ==
                RegKey.KeyStatus.Valid)
            {
                mnuManage.Enabled = true;
                this.Text += " (Registered to " +
                    configKey.KeyHolder + ")";
                mnuRegister.Enabled = false;
                mnuEnterKey.Enabled = false;
            }
        }
    }
    catch
    {
    }
}
```

The `ReadConfigFile()` method reads the contents of the specified configuration file. The XML content is stored in **readDoc** and the keys in the XML content are stored in **configKeys**. If the key for this application exists, it is validated. If the key is valid, full access is given to the application's functionality (in this case, the *Manage* menu item is enabled).

Note that the first three arguments for the `Validate()` function must be the same as those specified for the `Validate()` function used in the **frmEnterKey** form to check whether the entered licence key is valid. Notice that the fourth argument must be `False` when validating a key that the user has already used to register the application with.

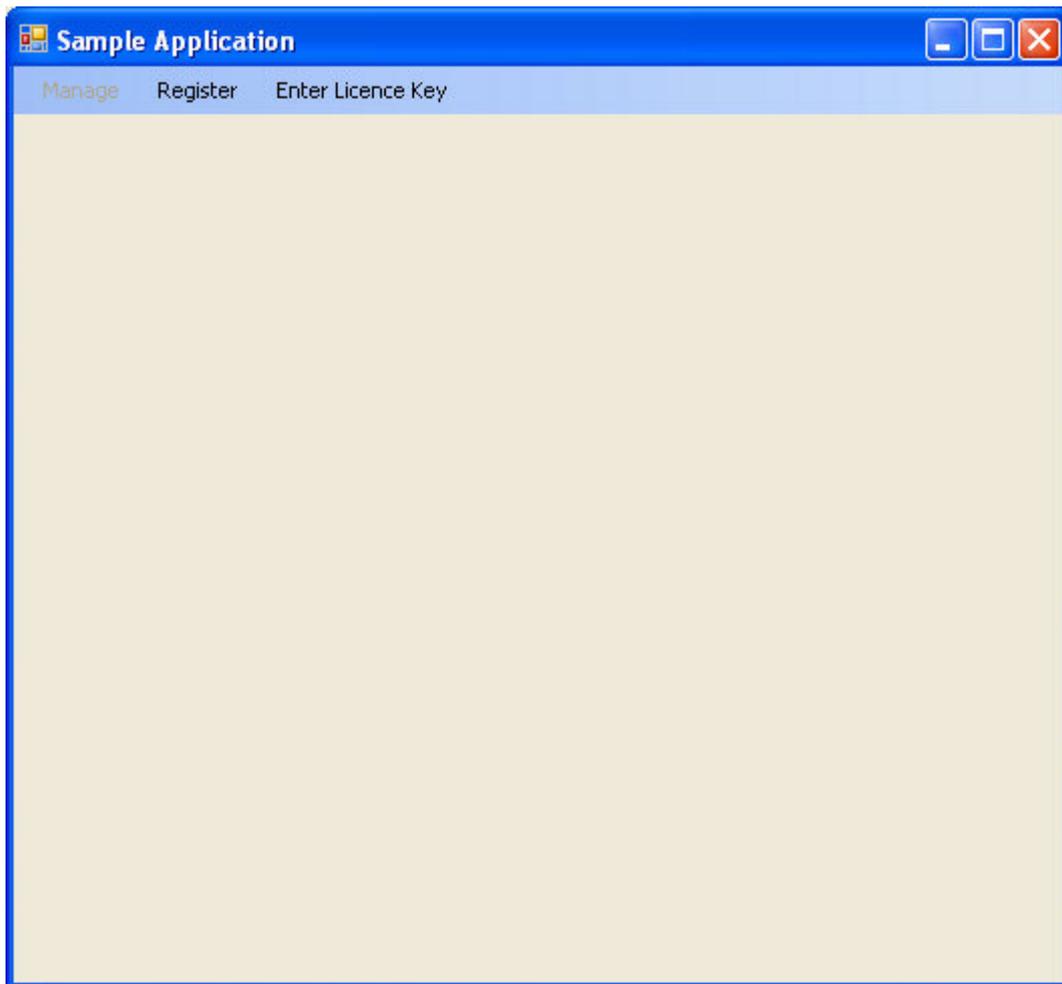
2. Add the `IDEquals()` function needed by the lists's `Find()` method in the code above to find the desired element in the list:

```
private static bool IDEquals(RegKey.configKey configKey)
{
    if (configKey.ID == "SampleAppKeyID")
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

2.9 Test the Application

To test the sample application, the steps that will be followed by a typical user who registers the application for the first time are now examined. Remember that if you are using the evaluation version of ZQure (i.e. you haven't registered ZQure yet), licence key validation will fail. To fully test the sample application, you should first register ZQure.

1. Run the sample application.



2. Click on the *Register* menu item to register the application, type in your details and then click the *Register* button.

Important: Your e-mail address will be used to send you your licence key.

*Name: Joe Soap

*E-mail Address: joe@mydomain.com

*Phone No: (123) 456 7890 Fax No:

*Address: 1 2nd Avenue
Thirstyville

*City: Port Fifth *Postal Code/Zip: 1234

*Country: South Africa

Company:

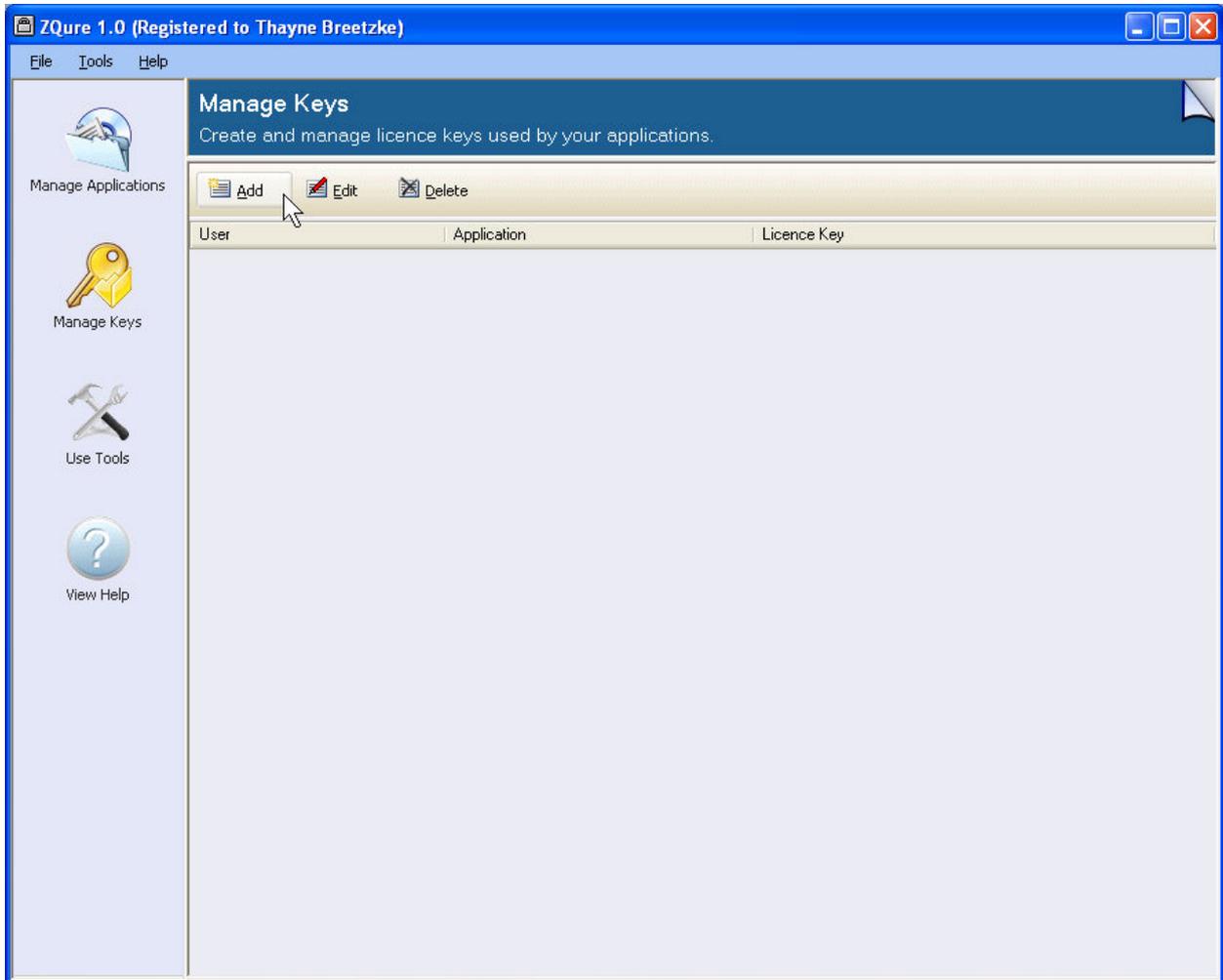
* = Required field

Register

Please e-mail the following text to the developer in order to receive your licence key:

```
<registration>
<application>ZQure Sample Application</application>
<username>Joe Soap</username>
<computerID>2DA64EB8A64C3A7F463C53F59F549735688F236A</computerID>
<applicationID>D7963AA13A9DFAA6394DEE53D955F254E956E94D</applicationID>
<email>joe@mydomain.com</email>
<phone>[123] 456 7890</phone>
<fax></fax>
<address1>1 2nd Avenue</address1>
<address2>Thirstyville</address2>
<city>Port Fifth</city>
<code>1234</code>
<country>South Africa</country>
<company></company>
</registration>
```

3. At this point the user will e-mail the resultant XML text to the developer in order to receive a licence key. Once received, the developer will copy the XML text to the clipboard and create a licence key using the ZQure manager. Therefore, start up the ZQure manager and click *Manage Keys*.



- Click on the *Add* button to add a new licence key.

Add Licence Key and User Details

Key Details | User Details | Import

*Username:

Computer ID:

Application ID: None Expiry Date:

*Licence Key:

For User:

* = Required field

- Because all the required registration details are in the XML text copied to the clipboard, click the *Import* tab, paste the text in the box and click the *Import Data* button. The form will be loaded with the registration details.

Add Licence Key and User Details

Key Details | User Details | Import

*Username: Joe Soap

Computer ID: 2DA64EB8A64C3A7F463C53F59F549735688F236A

Application ID: ZQure Sample Application Expiry Date:

*Licence Key:

For User:

* = Required field

- Click the *Generate Key* button to generate a licence key based on the form data.

Add Licence Key and User Details

Key Details | User Details | Import

*Username: Joe Soap

Computer ID: 2DA64EB8A64C3A7F463C53F59F549735688F236A

Application ID: ZQure Sample Application Expiry Date:

*Licence Key: 7AA9B9-CC4C76-6F8250-52173A-5622CB-D9CE24-83F868-909E90-0C95E9-801402-72F9F4-F43C8C-F46C3F-031DB0-18FA **Generate Key**

For User: Licence key details for ZQure Sample Application (please keep SAFE and SECRET):

 7AA9B9-CC4C76-6F8250-52173A-5622CB-D9CE24-83F868-909E90-0C95E9-801402-72F9F4-F43C8C-F46C3F-031DB0-18FA

* = Required field Save Cancel

The licence key is displayed as well as the following details of the key that the developer can e-mail to the user:

```

Licence key details for ZQure Sample Application (please keep SAFE and SECRET):
-----
7AA9B9-CC4C76-6F8250-52173A-5622CB-D9CE24-83F868-909E90-0C95E9-801402-72F9F4-
F43C8C-F46C3F-031DB0-18FA

User: Joe Soap
-----

This licence key is authorized for use by Joe Soap only and only on a computer having an ID of
2DA64EB8A64C3A7F463C53F59F549735688F236A. The key has no expiry date.

WARNING: Unauthorized use, reproduction or distribution of licence keys may result in severe
civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.
    
```

- Return to the sample application, and click the *Enter Licence Key* menu item. The user must then copy and paste the username and licence key that he/she received and click *OK*.

Enter Licence Key

Important: Copy your name and licence key from the registration document that you received.

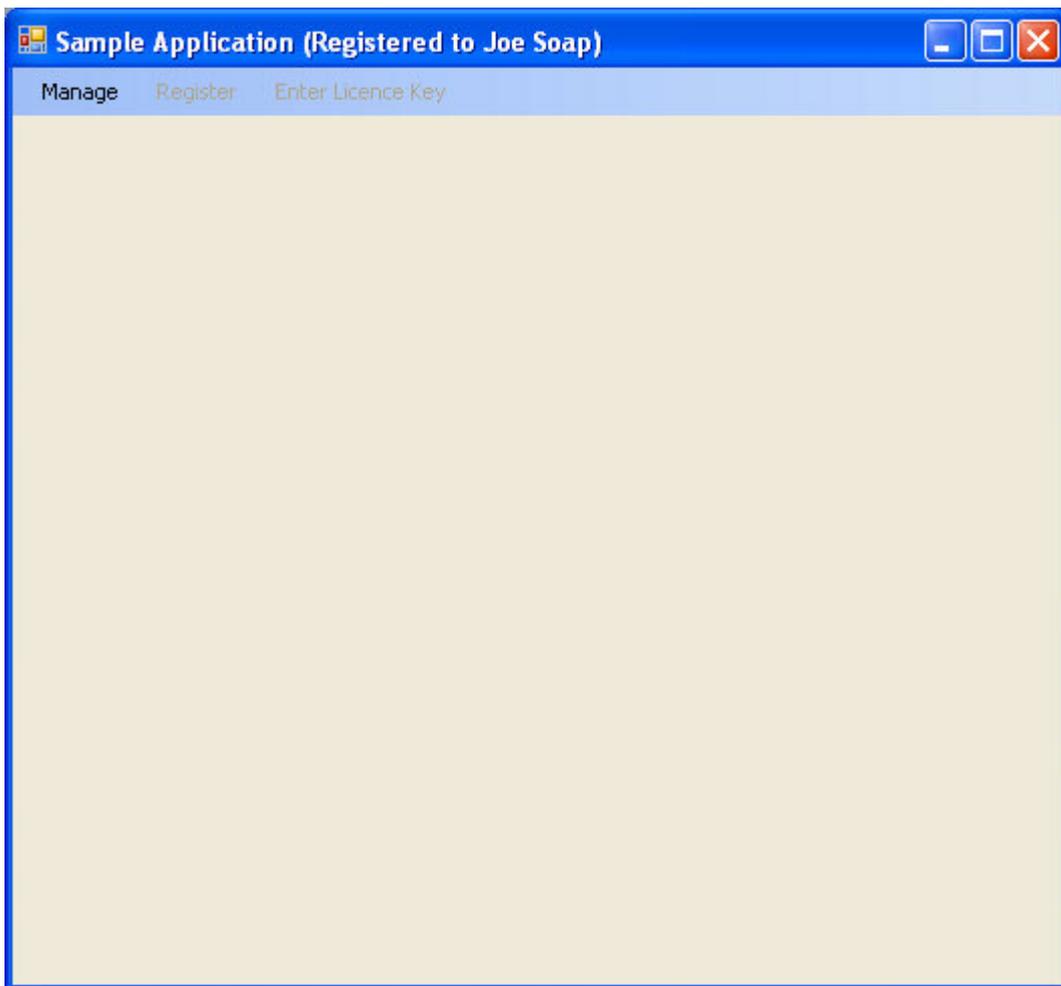
*Name: Joe Soap

*Licence Key: 7AA9B9-CC4C76-6F8250-52173A-5622CB-D9CE24-83F868-909E90-0C95E9-801402-72F9F4-F43C8C-F46C3F-031DB0-18FA

* = Required field

OK

- The *Manage* menu item then becomes available, as shown below:

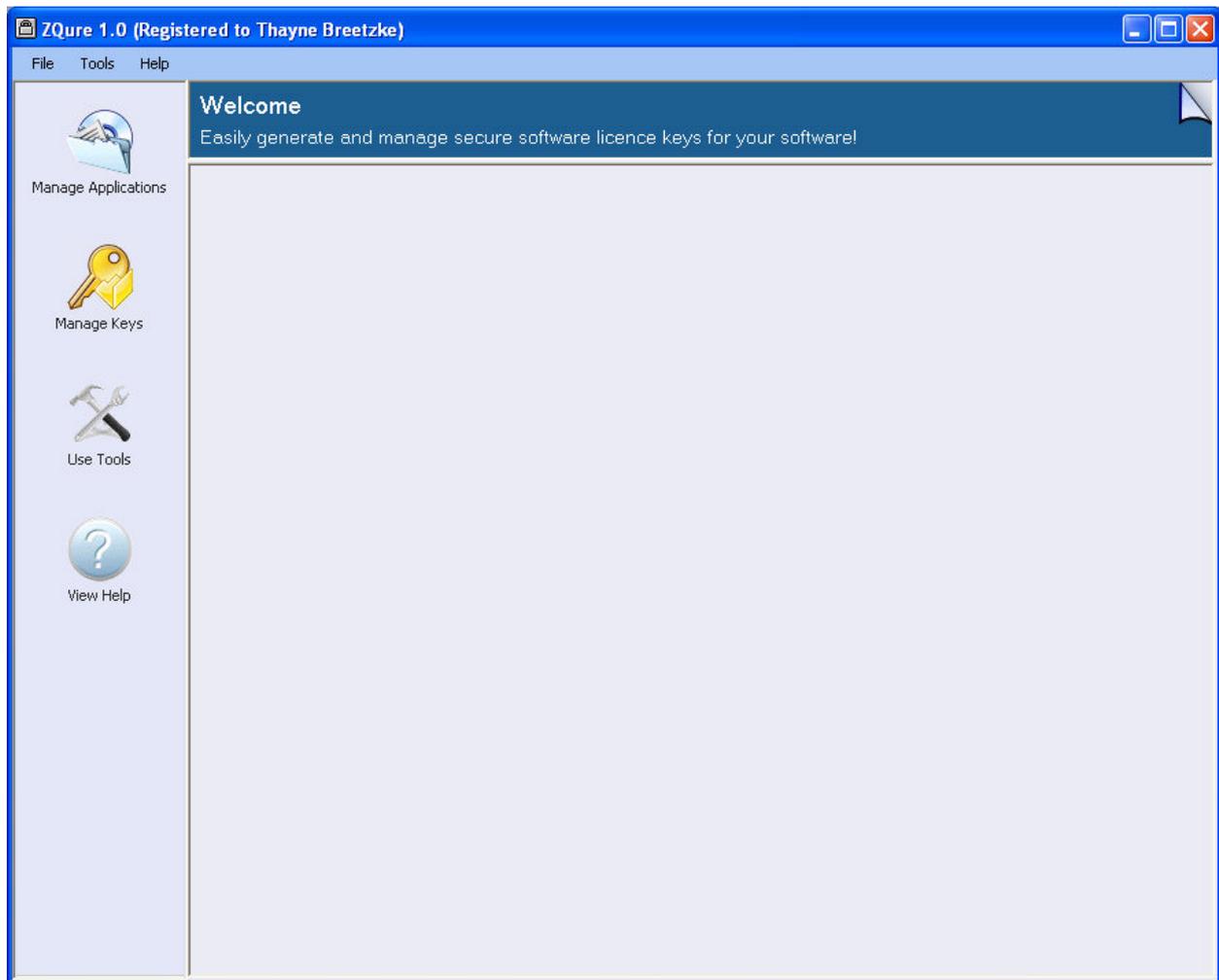


As shown in this example, it is recommended that the name that the user registered with is displayed in the application.

9. Close and re-start the sample application. The user still has access to the *Manage* menu item.

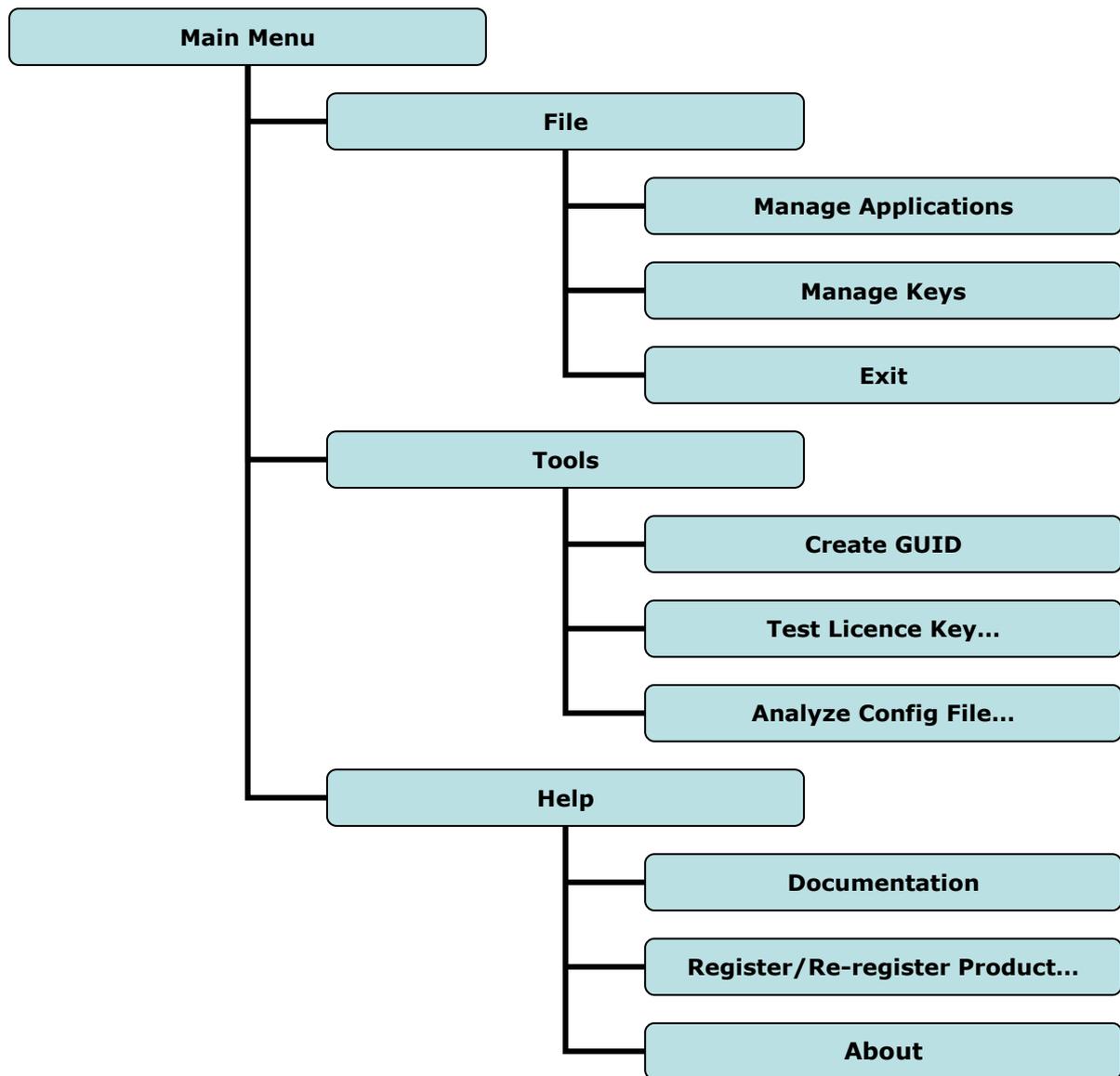
3 The ZQure Management Tool

The ZQure management tool is an essential part of the ZQure software licencing and protection system. It works together with the registration component to allow the developer to manage a list of applications for which licence keys can be generated, create licence keys to be issued to users, and to manage the licence keys issued to them. Its main screen is shown below together with an overview of the structure of the menu.



Commonly used menu items are available in the panel on the left of the manager window and are accessible with a single mouse click.

The user to which ZQure is registered is displayed in the title bar, unless an unregistered evaluation version of the software is running.



The ZQure Manager's Menu Structure

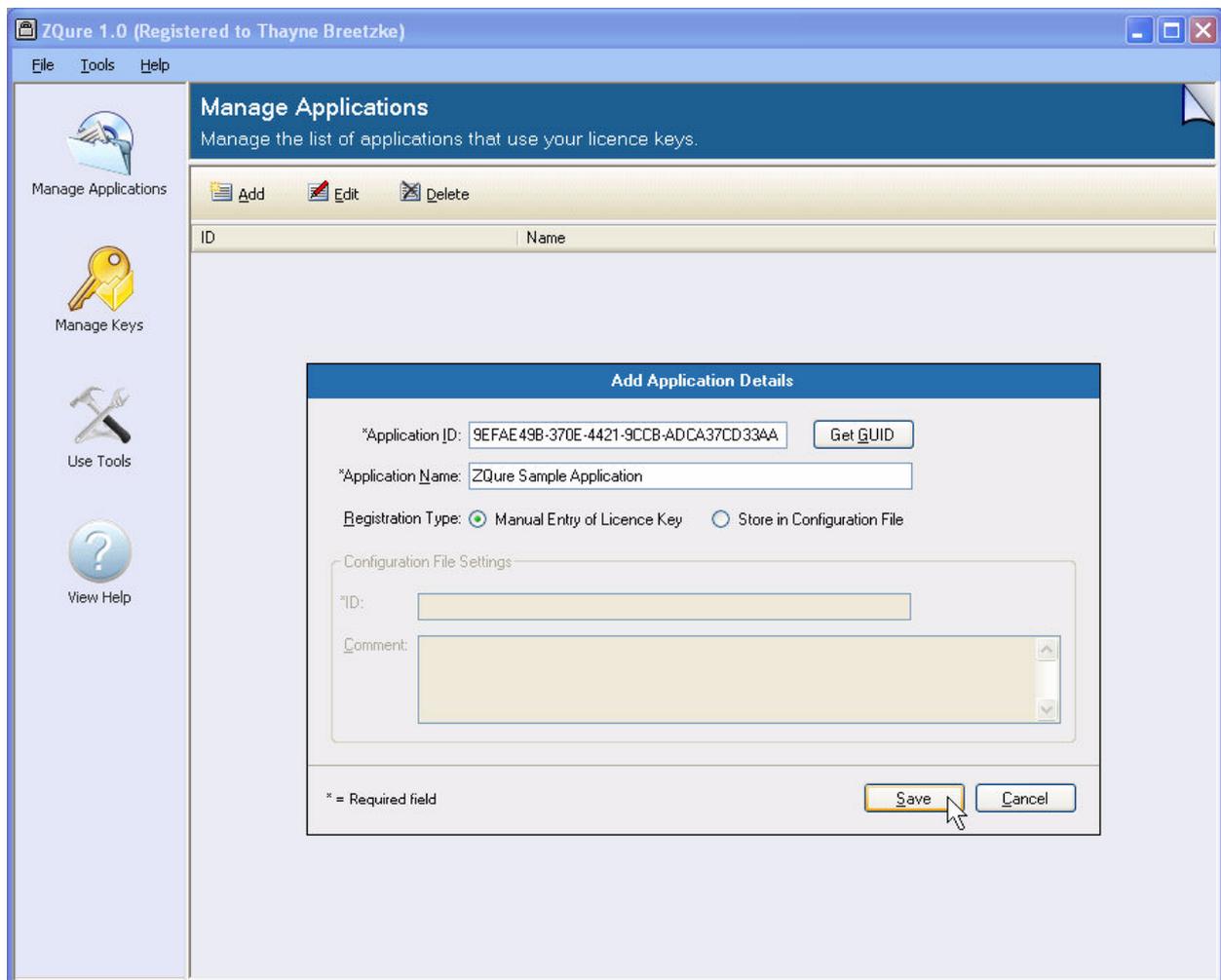
This section examines the various facilities available in the management tool as well as how to use them.

3.1 Managing Applications

To create a licence key that is based upon an application ID, a record must first be created to contain the details of that application. Simply click *Manage Applications* to see a list of existing application records. You can:

- **Add a new application.** Click the *Add* button or press the *Insert* key.
- **Edit an existing application.** Select the application to edit and click the *Edit* button or press the *Enter* key.
- **Delete an existing application.** Select the application to delete and click the *Delete* button or press the *Delete* key.

The first step to use ZQure in your software is usually to add the details of your application to the ZQure database, as shown below:



The following details are required for an application:

- **Application ID (required).** Each application must be given a string identifier, called an application ID. An application ID is automatically generated for a new application, but this default ID can be changed if necessary, although the default application ID is usually used. A new application ID can be generated by clicking the *Get GUID* button.
- **Application Name (required).** This is simply a descriptive name for the application.
- **Registration Type (required).** There are two ways that a user might register a copy of your software:
 - By manually entering a licence key generated by ZQure (default). This is usually used for a software application that has a user interface.
 - By including the configuration information generated by ZQure in the relevant software's configuration file. This is useful when the software does not have a user interface, such as a component that the user can use in his or her software. The user would simply copy the automatically generated configuration file information to software's configuration file (or create the configuration file, if it does not exist, and then paste the information into it). The file would contain a safe version of the licence key generated for the user and would have the following format:

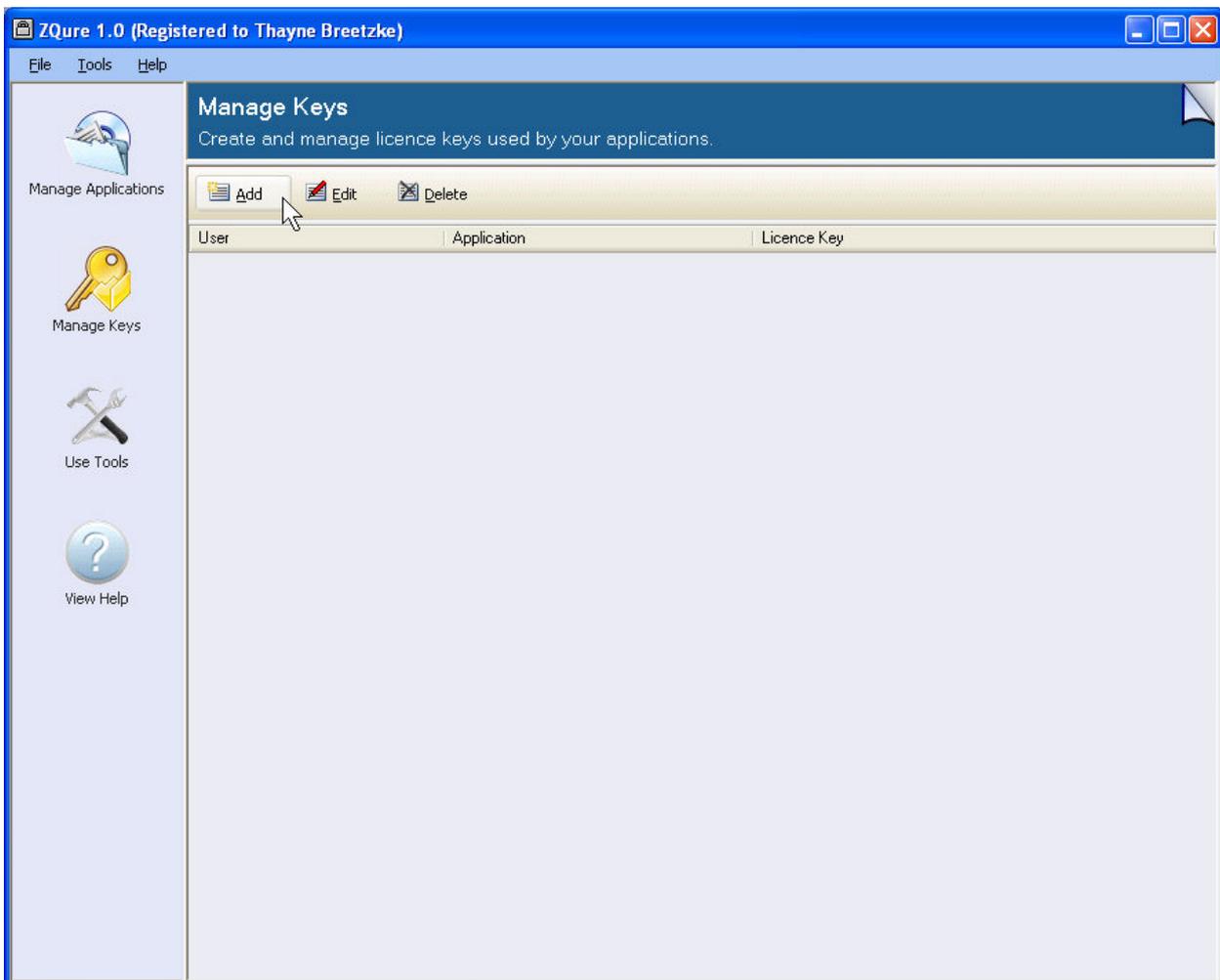
```
<configuration>
  <licenceKeys>
    <!-- Comment -->
    <key id="keyID" type="type" keyHolder="keyHolder" value="keyValue"
      data="keyData" />
  </licenceKeys>
  <!--Other configuration settings-->
</configuration>
```

If the *configuration file* registration type is chosen, an ID must be provided, along with an optional comment:

- **ID (required)**. This is a human-readable string identifier (e.g. ZQureSampleApp v1) used for the **id** property of the XML **key** element (see the above configuration file format). Note that it is not, and should not be, the same as the application ID used for the application.
- **Comment (optional)**. This is the comment text that is included in the configuration file information (see the above configuration file format). Two predefined comments are available, one of which can be selected by right-clicking the *Comment* textbox. These predefined comments are exposed as constants in the registration component. One comment is suitable for an executable software application and the other for a library/component. Refer to sections 4.11 and 4.12 for further information about these two constants.

3.2 Managing Licence Keys

Record is kept of each licence key that you generate for your applications. Simply click *Manage Keys* to see a list of the existing licence key records. You can add, edit and delete licence keys in the same way as you did to manage your applications.



The screenshot shows a dialog box titled "Add Licence Key and User Details". It features three tabs: "Key Details", "User Details", and "Import". The "Key Details" tab is active and contains the following fields:

- *Username: Joe Soap
- Computer ID: 2DA64EB8A64C3A7F463C53F59F549735688F236A
- Application ID: ZQure Sample Application (dropdown menu)
- Expiry Date: (empty date field)
- *Licence Key: (empty text box) with a "Generate Key" button
- For User: (empty list box)

At the bottom left, a note states "* = Required field". At the bottom right, there are "Save" and "Cancel" buttons.

Three tabs are present when adding and editing licence key details:

- **Key Details.** Details specific to the licence key and its creation.
- **User Details.** Details of who the licence key has been issued to (i.e. the key holder).
- **Import.** Allows you to easily and quickly import the key and user details from an XML string generated by your application and e-mailed to you by the user when he/she registers your software.

Key Details

The following details are required in order to create a licence key record:

- **Username (required).** The name of the user who the key has been issued to (i.e. the key holder).
- **Licence key (required).** The actual licence key string generated by ZQure after clicking the *Generate Key* button.

Depending on the chosen scope of the licence key (see Section 1.3 – *What is a Licence Key?*), you may need to supply values for each of the following before generating a licence key:

- **Computer ID (optional).** An optional identifier that ties the licence key to a specific physical entity, such as a computer or network.
- **Application ID (optional).** An optional identifier that ties the licence key to a specific application. For ease of use, the application ID is chosen by selecting the corresponding application name from the list box.
- **Expiry date (optional).** An optional expiry date for the licence key. If a date is not provided, the key will not expire.

Generally, the scope of a licence key is made as specific as possible by specifying a username, computer ID and application ID, and specifying an expiry date, if it is applicable. After you generate a licence key, details of the key are displayed in the *For User* textbox. This information is useful to include in an e-mail to the user.

The following is an example of the information generated in the *For User* textbox:

```

Licence key details for ZQure Sample Application (please keep SAFE and SECRET):
-----
7AA9B9-CC4C76-6F8250-52173A-5622CB-D9CE24-83F868-909E90-0C95E9-801402-72F9F4-
F43C8C-F46C3F-031DB0-18FA

User: Joe Soap
-----

This licence key is authorized for use by Joe Soap only and only on a computer having an ID of
2DA64EB8A64C3A7F463C53F59F549735688F236A. The key has no expiry date.

WARNING: Unauthorized use, reproduction or distribution of licence keys may result in severe
civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.
    
```

User Details

The following details of the user of the licence key can be stored (note that all fields are optional):

- E-mail address
- Phone number
- Fax number
- Address
- City
- Postal code
- Country
- Company

Import

When a user registers his/her copy of your software, your application can use the [RegInfoToXML](#) method in the registration component to generate an XML string containing the user's details, as entered on the registration form. The user can then e-mail this string to you and you can use the *Import* tab to paste the XML string and import the user data in one easy step. The fields in the *Key Details* and *User Details* tabs will be filled with the values in the XML string, thus saving you from capturing the details manually. The imported data can be changed as needed after the import, but before generating a key.

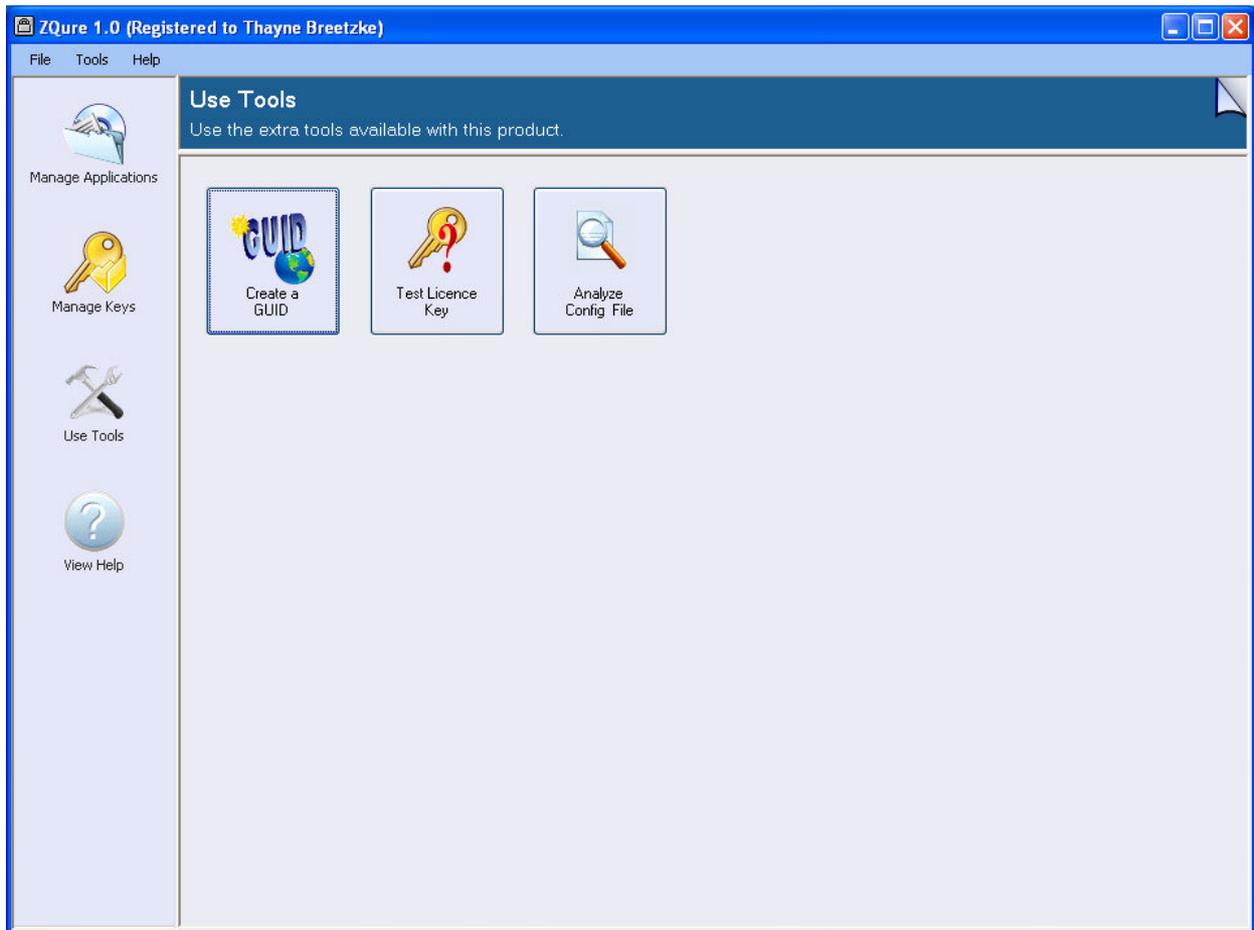
PLEASE NOTE

The evaluation version of ZQure generates invalid licence keys. Please register ZQure to generate valid licence keys.

3.3 Using the Tools

The following tools are available in the *Tools* menu or by clicking *Use Tools* in the side panel:

- **Create a GUID.** This tool can be used to generate a GUID for use elsewhere in the manager or in your application.
- **Test licence key.** To test the validity of a licence key, simply type in its details on the form and click the *Test Key* button. The result of the test is then displayed.
- **Analyze config file.** To analyze the licence key(s) present in a configuration file, simply select the configuration file after clicking the *Select File* button. The results of the analysis of the selected file are immediately displayed and may be useful for diagnostic purposes.



3.4 Viewing Help

Full on-line documentation is provided and accessible via the menu or side panel. Adobe Acrobat Reader is required to read the documentation.

3.5 Registering ZQure

Registering ZQure is simple and follows the typical steps your users might follow to register your own applications. To register ZQure, click *Register Product...* on the main menu (or *Re-register Product...* if the software has previously been registered), or the *Register...* (or *Re-register...*) button on the *About* screen. Complete the second part of the form and click the *Next* button. This step is illustrated below:

Register Product

If you have already purchased a licence key:

Important: Copy your name and licence key from the registration document that you received.

*Name:

*Licence Key:

If you have not yet purchased a licence key:

Important: Your e-mail address will be used to send you your licence key.

*Name:

*E-mail Address:

*Phone No: Fax No:

*Address:

*City: *Postal Code/Zip:

*Country:

*Company:

* = Required field

After clicking *Next*, your registration information will be displayed on the form in XML format.

Register Product

The registration information shown below has been copied to the clipboard. Please paste it in an e-mail and then send it to SophicEdge@mweb.co.za. To receive your licence key, please make a payment of R299 into the following bank account:

<p>Bank: ABSA Branch: 632005 Account: 9062548285 Type: Savings</p>	<p>Please Note:</p> <p>Please use your telephone number (the same one that you registered with) as the reference number for your deposit.</p>
--	--

As soon as your registration details and payment are received, your licence key will be e-mailed to you.

```
<registration>
<application>ZQure 1.0</application>
<username>Joe Soap</username>
<computerID>2DA64EB8A64C3A7F463C53F59F549735688F236A</computerID>
<applicationID>BC670E62016F006AF571146BF055F254E956E94D</applicationID>
<email>joe@mydomain.com</email>
<phone>[123] 456 7890</phone>
<fax></fax>
<address1>1 2nd Avenue</address1>
<address2>Thirstyville</address2>
<city>Port Fifth</city>
<code>1234</code>
<country>South Africa</country>
<company></company>
</registration>
```

As instructed, paste the XML information into an e-mail and e-mail it to the e-mail address shown. To receive your licence key, make a payment of the required registration fee using the banking details shown and using your telephone number as the reference number for your deposit. Upon receipt of your registration details and payment, your licence key will be e-mailed to the address you provided in the registration process.

When you receive your licence key, return to the *Registration* form and fill out the details in the first part of the form.

Register Product

If you have already purchased a licence key:

Important: Copy your name and licence key from the registration document that you received.

*Name:

*Licence Key:

If you have not yet purchased a licence key:

Important: Your e-mail address will be used to send you your licence key.

*Name:

*E-mail Address:

*Phone No: Fax No:

*Address:

*City: *Postal Code/Zip:

*Country:

Company:

* = Required field

Click the *OK* button to complete the registration process. Please note that the name you enter must be exactly the same as the one you registered with and the licence key must be exactly as shown in the e-mail. Therefore, it is best to copy these values directly from the e-mail you received and paste them on the form.

PLEASE NOTE

The ZQure application folder contains the **Library** and **Sample Application** folders. After successful registration, the **Library** folder will contain the **registration.dll** component and the **registration.dll.config** file. Both files are necessary to validate licence keys in your own software.

4 The ZQure Registration Library

The ZQure registration library is a .DLL that is used by the developer's application to perform tasks such as validating licence keys, obtaining a unique computer identifier, and reading and writing configuration file data. The library contains one class, [RegKey](#), which exposes the following members:

Constructors

Name	Description
RegKey	Initializes a new instance of the RegKey class.

Methods

Static	Name	Description
✓	CreateConfigDoc	Creates an XML configuration document based on data received in an <i>XmlDocument</i> argument and the licence keys received in a <i>List of configKey</i> elements.
✓	GetComputerID	Returns an ID unique to the computer executing the method.
✓	ReadConfigFile	Reads an XML configuration file and stores the content in an <i>XmlDocument</i> instance. Stores the keys that appear in the <code><licenceKeys></code> section in a <i>List of configKey</i> elements.
✓	RegInfoFromXML	Converts a given XML string to the original registration information strings. The encrypted application ID is automatically decrypted.
✓	RegInfoToXML	Converts given registration information strings to an XML string. The application ID is automatically encrypted for safety.
	ToString	Returns a safe string version of a licence key instance for publishing in a database, configuration file, etc.
	Validate	Checks whether the value of a licence key instance is valid.
✓	WriteConfigFile	Writes an XML configuration file based on data received in an <i>XmlDocument</i> argument and the licence keys received in a <i>List of configKey</i> elements.

Properties

Name	Description
Value	Gets/sets the value of a licence key instance.

Constants

Name	Description
<code>defaultConfigAppComment</code>	The default comment for licence keys that pertain to applications.
<code>defaultConfigLibComment</code>	The default comment for licence keys that pertain to libraries.

Enumerated Types

Name	Description
<code>KeyStatus</code>	Specifies the result of a licence key validation check using the <code>Validate</code> method. Possible values: <ul style="list-style-type: none"> • Valid. The key is valid. • Expired. The key has expired. • Invalid. The key is invalid. • ConfigError. There is an error in or with the required configuration file. • Error. An unlikely cryptographic exception occurred.

Structures

Name	Description
<code>keyStruct</code>	Represents key-related data (protected; used internally).
<code>userStruct</code>	Represents user-related data.

Classes

Name	Description
<code>configKey</code>	Represents a licence key in a configuration file.

Each of these members is examined further in the sections that follow.

PLEASE NOTE

To use the library in your software, carefully follow these steps:

1. Add a reference to the **Registration.dll** component in the **Library** folder.
2. If you have registered ZQure, there will be a **Registration.dll.config** file in the **Library** folder. Add it to the solution (right click *Sample Application* | *Add* | *Existing Item...*) and set its *Copy to Output Directory* property to *Copy Always*. Be sure that the **Registration.dll** and **Registration.dll.config** files are copied to the output folder after building your project. If you do not add the configuration file, or if you are using the evaluation version of ZQure, the **Registration** component will not be able to validate licence keys.

4.1 RegKey Constructor

Initializes a new instance of the [RegKey](#) class.

Syntax

```
public RegKey(string key)
```

Parameters

key

Type: string

The key value.

Remarks

key can be either a secret licence key string as generated by the ZQure management tool, or a safe version of the key, as returned by [ToString](#).

Example

The following example creates a licence key instance based upon the key string entered in a textbox:

```
RegKey key = new RegKey(txtLicenceKey.Text.Trim());
```

See Also

[ToString](#)

[Validate](#)

[Value](#)

4.2 CreateConfigDoc Method

Creates an XML configuration document based on data received in an *XmlDocument* argument and the licence keys received in a *List* of *configKey* elements.

Syntax

```
public static XmlDocument CreateConfigDoc(XmlDocument doc,
                                         List<configKey> keys)
```

Parameters

doc

Type: XmlDocument

The variable which contains the configuration file content to be included in the resultant document. To create a new document containing default content, pass a *null* argument.

keys

Type: List<configKeys>

The *List* variable which contains the list of keys to be included in the <licenceKeys> section of the configuration document.

Return Value

Type: XmlDocument

The XML document created.

Exceptions

Exception	Condition
<code>ApplicationException</code>	Root element must be <configuration>.

Remarks

The licence keys are grouped within a <licenceKeys> element which, in turn, is nested within a <configuration> root element. Each licence key is defined by a <key> element in the format shown below:

```
<configuration>
  <licenceKeys>
    <!-- Comment -->
    <key id="keyID" type="type" keyHolder="keyHolder" value="keyValue"
      data="keyData" />
  </licenceKeys>
  <!--Other configuration settings-->
</configuration>
```

Note that all occurrences of the *&keyHolder;* placeholder in each key comment are replaced with the value of the *keyHolder* property for that corresponding key.

Example

The following example, creates a configuration file containing one licence key.

```
List<RegKey.configKey> configKeys = new List<RegKey.configKey>();
RegKey.configKey configKey = new RegKey.configKey();
configKey.ID = "SampleAppKeyID";
configKey.Type = "Application";
configKey.KeyHolder = "Joe Soap";
configKey.Value = key.ToString();
configKey.Data = "";
configKey.Comment = RegKey.defaultConfigAppComment;

configKeys.Add(configKey);
try
{
    XmlDocument doc = RegKey.CreateConfigDoc(null, configKeys);
    :
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK);
}
```

See Also

[configKeys](#)
[defaultConfigAppComment](#)
[defaultConfigLibComment](#)

4.3 GetComputerID Method

Returns an ID unique to the computer executing the method.

Syntax

```
public static string GetComputerID()
```

Return Value

Type: string
The computer ID.

Remarks

The computer ID is based upon CPU, hard drive and BIOS information. Note that if any of these physical components are replaced then this is considered a major hardware change and the ID generated by this method will change.

See Also

[Validate](#)

4.4 ReadConfigFile Method

Reads an XML configuration file and stores the content in an *XmlDocument* instance. Stores the keys that appear in the `<licenceKeys>` section in a *List of configKey* elements.

Syntax

```
public static void ReadConfigFile(string path,
                                out XmlDocument doc,
                                out List<configKey> keys)
```

Parameters

path

Type: string

The full path to the configuration file (including the file name).

doc

Type: XmlDocument

The variable to receive the configuration file content.

keys

Type: List<configKeys>

The *List* variable to receive the keys in the `<licencKeys>` section of the configuration file.

Exceptions

Exception	Condition
<code>ArgumentException</code>	<i>path</i> is an empty string ("").
<code>ArgumentNullException</code>	<i>path</i> is a null reference (Nothing in Visual Basic).
<code>FileNotFoundException</code>	The file cannot be found.
<code>DirectoryNotFoundException</code>	The specified path is invalid, such as being on an unmapped drive.
<code>IOException</code>	<i>path</i> includes an incorrect or invalid syntax for file name, directory name, or volume label.
<code>XmlException</code>	There is a load or parse error in the XML. In this case, the document remains empty.

Remarks

The licence keys must be grouped within a `<licenceKeys>` element which, in turn, must be nested within a `<configuration>` root element. Each licence key must be defined by a `<key>` element in the format shown below:

```
<configuration>
  <licenceKeys>
    <!-- Comment -->
    <key id="keyID" type="type" keyHolder="keyHolder" value="keyValue"
      data="keyData" />
  </licenceKeys>
  <!--Other configuration settings-->
</configuration>
```

Example

The following example reads an application's configuration file, finds the key defined by the predicate (not shown), and creates a licence key instance based upon the licence key string.

```
try
{
    XmlDocument readDoc = new XmlDocument();
    List<RegKey.configKey> configKeys = new List<RegKey.configKey>();
    RegKey.ReadConfigFile(Application.ExecutablePath + ".config", out readDoc,
    out configKeys);

    RegKey.configKey configKey = configKeys.Find(IDEquals);
    if (configKey != null)
    {
        RegKey key = new RegKey(configKey.Value);
        :
    }
    catch
    {
        MessageBox.Show(errorMessage, "Error", MessageBoxButtons.OK);
    }
}
```

See Also

[WriteConfigFile](#)
[configKeys](#)

4.5 *RegInfoFromXML* Method

Converts a given XML string to the original registration information strings. The encrypted application ID is automatically decrypted.

Syntax

```
public static bool RegInfoFromXML(string regInfoXML,
                                out userStruct data,
                                out XmlException error)
```

Parameters

regInfoXML
Type: string
The registration information in XML format.

data
Type: userStruct
The userStruct instance that will receive the registration information.

error
Type: XmlException
The XmlException instance that contains detailed information about an exception.

Return Value

Type: bool
Whether the XML string was successfully converted.

Remarks

When a user registers an application the application will typically encapsulate the registration information in an XML string for ease of transfer to the developer (see the [RegInfoToXML](#) method). [RegInfoFromXML](#) converts the XML string back to the original registration information strings for processing by the ZQure management tool. Note that the application ID is automatically decrypted.

Example

```
RegKey.userStruct data;  
XmlException error;  
  
if (RegKey.RegInfoFromXML(txtData.Text, out data, out error))  
{  
    if (data.usernameHasValue)  
        txtUsername.Text = data.username;  
    :  
}  
else  
    MessageBox.Show(error.Message, "Error", MessageBoxButtons.OK);  
}
```

See Also

[RegInfoToXML](#)
[userStruct](#)

4.6 RegInfoToXML Method

Converts the given registration information strings to an XML string. The application ID is automatically encrypted for safety.

Syntax

```
public static string RegInfoToXML(userStruct data)
```

Parameters

data

Type: [userStruct](#)

The [userStruct](#) instance that contains the registration information.

Return Value

Type: string

The registration information in XML format.

Remarks

When a user registers an application the application will typically encapsulate the registration information in an XML string for ease of transfer to the developer by using the [RegInfoToXML](#) method. Note that the application ID is automatically encrypted for safety.

Example

```
txtRegInfo.Text = RegKey.RegInfoToXML(new RegKey.userStruct(  
    "ZQure Sample Application",  
    txtName.Text, RegKey.GetComputerID(),  
    "9EFAE49B-370E-4421-9CCB-ADCA37CD33AA",  
    txtEmailAddr.Text, txtPhoneNo.Text, txtFaxNo.Text,  
    txtAddrLine1.Text, txtAddrLine2.Text, txtCity.Text,  
    txtPostalCodeZip.Text, txtCountry.Text, txtCompany.Text));
```

See Also

[RegInfoFromXML](#)
[userStruct](#)

4.7 ToString Method

Returns a safe string version of a licence key instance for publishing in a database, configuration file, etc.

Syntax

```
public string ToString()
```

Return Value

Type: string
The safe string version of the licence key.

Remarks

A licence key string will typically be generated by the ZQure management tool and e-mailed to the user who the key is registered to. This key must be kept safe and secret. When storing the key string in an application's database, configuration file, etc, it must be stored as a safe version of the key. This safe version is obtained by using [ToString](#).

Note that registering using a safe version of a licence key will fail. Furthermore, if the licence key is malformed or corrupt, the method returns the key string in its original form (i.e. unchanged).

Example

The following example creates a configuration file containing one licence key. The key value stored in the configuration file is stored in its safe format by using [ToString](#).

```
List<RegKey.configKey> configKeys = new List<RegKey.configKey>();  
RegKey.configKey configKey = new RegKey.configKey();  
configKey.ID = "SampleAppKeyID";  
configKey.Type = "Application";  
configKey.KeyHolder = username;  
configKey.Value = key.ToString();  
configKey.Data = "";  
configKey.Comment = RegKey.defaultConfigAppComment;  
  
configKeys.Add(configKey);
```

```
try
{
    RegKey.WriteConfigFile(Application.ExecutablePath + ".config", null,
        configKeys);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK);
}
```

See Also

Value

4.8 Validate Method

Checks whether the value of a licence key instance is valid.

Syntax

```
public KeyStatus Validate(string userName,
                          string computerID,
                          string applicationID,
                          bool isRegistering)
```

Parameters

userName

Type: string

The key holder's name.

computerID

Type: string

The identifier that ties the licence key to a specific physical entity, such as a computer or network and that was used when creating the key.

applicationID

Type: string

The identifier that ties the licence key to a specific application and that was used when creating the key.

isRegistering

Type: bool

True if registering with the key or *False* if checking whether a key is still valid.

Return Value

Type: KeyStatus

The result of the validation check.

Remarks

This method is involved in two important tasks in an application:

- **Registering the application.** This usually happens once. Thereafter, the safe version of the licence key used to register the application is stored with the application for future use.
- **Checking whether the licence key that was used to register the application is still valid.** This usually happens each time the application starts. The safe version of

the licence key is retrieved and validated. A licence key might become invalid if it has expired or if the licence key is restricted to a certain computer, but the software has been copied to another computer.

Example

The following example checks whether the licence key entered is valid for registering with:

```
//Create a licence key instance for the key string entered in the textbox
RegKey key = new RegKey(txtLicenceKey.Text.Trim());

//Validate the key
switch (key.Validate(txtRegisteredName.Text.Trim(),
                    RegKey.GetComputerID(),
                    "9EFAE49B-370E-4421-9CCB-ADCA37CD33AA", true))
{
    case RegKey.KeyStatus.Valid:
        //Process valid registration
        :
        break;
    case RegKey.KeyStatus.Expired:
        //User registered with valid, but expired, key.
        :
        break;
    case RegKey.KeyStatus.ConfigError:
        //Required configuration file is not present or valid.
        :
        break;
    case RegKey.KeyStatus.Error:
        //Unexpected error occurred.
        :
        break;
    default:
        //User registered with invalid key.
        :
        break;
}
```

See Also

[GetComputerID](#)
[KeyStatus](#)

4.9 WriteConfigFile Method

Writes an XML configuration file based on data received in an *XmlDocument* argument and the licence keys received in a *List* of *configKey* elements.

Syntax

```
public static void WriteConfigFile(string path,
                                   XmlDocument doc,
                                   List<configKey> keys)
```

Parameters

path

Type: string

The full path to the configuration file (including the file name).

doc

Type: XmlDocument

The variable which contains the configuration file content to be written. To create a new file containing default content, pass a *null* argument.

keys

Type: List<configKeys>

The *List* variable which contains the list of keys to be included in the <licenceKeys> section of the configuration file.

Exceptions

Exception	Condition
ApplicationException	Root element must be <configuration>.
ArgumentException	The encoding is not supported; the filename is empty, contains only white space, or contains one or more invalid characters.
UnauthorizedAccessException	Access is denied.
ArgumentNullException	<i>path</i> is a null reference (Nothing in Visual Basic).
DirectoryNotFoundException	The specified path is invalid, such as being on an unmapped drive.
IOException	<i>path</i> includes an incorrect or invalid syntax for file name, directory name, or volume label.

Remarks

The licence keys are grouped within a <licenceKeys> element which, in turn, is nested within a <configuration> root element. Each licence key is defined by a <key> element in the format shown below:

```
<configuration>
  <licenceKeys>
    <!-- Comment -->
    <key id="keyID" type="type" keyHolder="keyHolder" value="keyValue"
      data="keyData" />
  </licenceKeys>
  <!--Other configuration settings-->
</configuration>
```

Note that all occurrences of the &keyHolder; placeholder in each key comment are replaced with the value of the keyHolder property for that corresponding key.

Example

The following example creates a configuration file containing one licence key.

```
List<RegKey.configKey> configKeys = new List<RegKey.configKey>();
RegKey.configKey configKey = new RegKey.configKey();
configKey.ID = "SampleAppKeyID";
configKey.Type = "Application";
configKey.KeyHolder = username;
configKey.Value = key.ToString();
configKey.Data = "";
configKey.Comment = RegKey.defaultConfigAppComment;

configKeys.Add(configKey);
try
{
    RegKey.WriteConfigFile(Application.ExecutablePath + ".config", null,
        configKeys);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK);
}
```

See Also

[ReadConfigFile](#)
[configKeys](#)
[defaultConfigAppComment](#)
[defaultConfigLibComment](#)

4.10 Value Property

Gets/sets the value of a licence key instance.

Syntax

```
public string Value
```

Property Value

Type: string
The specified key value for the instance.

Remarks

Although the [RegKey](#) constructor is usually used to create a licence key instance based upon a licence key value, this property can be used to retrieve or change the value of the instance if necessary. Note that the property returns the licence key value in its secret form, not in its safe form. Therefore, this property should not be used to retrieve and store a licence key value in a database, configuration file, etc. Use [ToString](#) instead to retrieve the safe version of a key.

Example

The following example create a licence key instance containing no key value and then changes the licence key value to the key string entered in a textbox:

```
RegKey key = new RegKey("");  
key.Value = txtLicenceKey.Text.Trim();
```

See Also

[ToString](#)

4.11 *defaultConfigAppCommentConstant*

The default configuration file comment for licence keys that pertain to applications.

Remarks

This constant contains the following value:

```
Key holder: &keyHolder;
```

If you, or the person in charge of this software, are not the key holder named above, this software is being used ILLEGALLY. Unauthorized use, reproduction or distribution of this key or this software, or any portion of it, will be prosecuted to the maximum extent possible under the law.

When used as the comment for a `configKey` instance and the key is written to a configuration file using `WriteConfigFile`, the `&keyHolder;` placeholder is replaced with the value of the key's `KeyHolder` property.

See Also

[WriteConfigFile](#)
[configKeys](#)
[defaultConfigLibComment](#)
[configKey](#)

4.12 *defaultConfigLibComment Constant*

The default configuration file comment for licence keys that pertain to libraries.

Remarks

This constant contains the following value:

```
Key holder: &keyHolder;
```

The following library used by this software was purchased by the key holder named above. If you purchased or obtained this software from a person or company other than this key holder, this software is being used ILLEGALLY. Unauthorized use, reproduction or distribution of this key or this library, or any portion of it, will be prosecuted to the maximum extent possible under the law.

When used as the comment for a `configKey` instance and the key is written to a configuration file using `WriteConfigFile`, the `&keyHolder;` placeholder is replaced with the value of the key's `KeyHolder` property.

See Also

`WriteConfigFile`
`configKeys`
`defaultConfigAppComment`
`configKey`

4.13 *KeyStatus* Enumerated Type

Specifies the result of a key validation check using the `Validate` method.

Syntax

```
public enum KeyStatus
```

Members

Member Name	Description
Valid	The key is valid.
Expired	The key has expired.
Invalid	The key is invalid.
ConfigError	There is an error in or with the required configuration file.
Error	An unlikely cryptographic exception occurred.

Remarks

Used for the result of the `Validate` method.

See Also

`Validate`

4.14 *keyStruct* Structure

Represents key-related data (protected; used internally).

Syntax

```
[Serializable()]  
protected struct keyStruct
```

Constructors

```
public keyStruct(string name,
                DateTime expiryDate,
                string computerID,
                string applicationID)
```

Initializes a new instance of `userStruct` using the given values.

Remarks

This structure is used internally and is not intended to be used directly from your code.

4.15 *userStruct* Structure

Represents user-related data.

Syntax

```
[Serializable()]
public struct userStruct
```

Constructors

```
public userStruct()
```

Initializes a new instance of `userStruct` using the default values.

```
public userStruct(string applicationName,
                 string username,
                 string computerID,
                 string applicationID,
                 string email,
                 string phone,
                 string fax,
                 string address1,
                 string address2,
                 string city,
                 string code,
                 string country,
                 string company)
```

Initializes a new instance of `userStruct` using the default values.

Properties

applicationName

Type: string

The name of the application.

applicationNameHasValue

Type: bool

Whether *applicationName* has been assigned a value.

username

Type: string

The name of the user.

usernameHasValue

Type: bool

Whether *username* has been assigned a value.

computerID

Type: string

The *computerID* value.

computerIDHasValue

Type: bool

Whether *computerID* has been assigned a value.

applicationID

Type: string

The *applicationID* value.

applicationIDHasValue

Type: bool

Whether *applicationID* has been assigned a value.

email

Type: string

The e-mail address of the user.

emailHasValue

Type: bool

Whether *email* has been assigned a value.

phone

Type: string

The phone number of the user.

phoneHasValue

Type: bool

Whether *phone* has been assigned a value.

fax

Type: string

The fax number of the user.

faxHasValue

Type: bool

Whether *fax* has been assigned a value.

address1

Type: string

The first line of the user's address.

address1HasValue

Type: bool

Whether *address1* has been assigned a value.

address2

Type: string

The second line of the user's address.

address2HasValue

Type: bool

Whether *address2* has been assigned a value.

city

Type: string

The user's city.

cityHasValue

Type: bool

Whether *city* has been assigned a value.

code

Type: string

The user's postal code.

codeHasValue

Type: bool

Whether *code* has been assigned a value.

country

Type: string

The user's country.

countryHasValue

Type: bool

Whether *country* has been assigned a value.

company

Type: string

The user's company.

companyHasValue

Type: bool

Whether *company* has been assigned a value.

Remarks

Used by the [RegInfoFromXML](#) method to return user data read from an XML string.

Example

```
RegKey.userStruct data;  
string errorMessage;  
  
if (RegKey.RegInfoFromXML(txtData.Text, out data, out errorMessage))  
{  
    if (data.usernameHasValue)  
        txtUsername.Text = data.username;  
    :  
}  
else  
    MessageBox.Show(errorMessage, "Error", MessageBoxButtons.OK);
```

See Also

[RegInfoFromXML](#)

4.16 *configKey* Class

Represents a licence key in a configuration file.

Syntax

```
public class configKey
```

Constructors

```
public configKey()
```

Initializes a new instance of *configKey* using the default values.

```
public configKey(string id,  
                string type,  
                string keyHolder,  
                string value,  
                string data,  
                string comment)
```

Initializes a new instance of *configKey* using the given values.

Properties

ID

Type: string

The configuration file key ID for the licence key element.

Type

Type: string

The type of licence key element, either "Application" or "Library".

KeyHolder

Type: string

The name of the key holder of the licence key (i.e. the name that the user registered as).

Value

Type: string

The licence key's value (i.e. the safe version of the licence key retrieved using *ToString*).

Data

Type: string

An optional data string associated with the licence key.

Comment

Type: string

The comment for the licence key element.

Remarks

Used by the *ReadConfigFile* and *WriteConfigFile* methods to read and write licence key information to/from a configuration file.

Example

The following example creates a configuration file containing one licence key.

```
List<RegKey.configKey> configKeys = new List<RegKey.configKey>();
RegKey.configKey configKey = new RegKey.configKey();
configKey.ID = "SampleAppKeyID";
configKey.Type = "Application";
configKey.KeyHolder = username;
configKey.Value = key.ToString();
configKey.Data = "";
configKey.Comment = RegKey.defaultConfigAppComment;

configKeys.Add(configKey);
try
{
    RegKey.WriteConfigFile(Application.ExecutablePath + ".config", null,
        configKeys);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK);
}
```

See Also

[ReadConfigFile](#)
[ToString](#)
[WriteConfigFile](#)
[defaultConfigAppComment](#)
[defaultConfigLibComment](#)

5 Tips for Developers

The following tips are useful to developers to help ensure that their software remains protected when using ZQure:

- **Ensure the visibility of the registered user's name.** Once your software application has been registered to a user, the user's name should be clearly visible and obvious somewhere within the application. An ideal location for the name is in the title bar of the application's main form. This will help discourage pirate users from attempting to copy and use the software.
- **Use unrecognizable application IDs.** When using the ZQure manager to add a new application, the application ID defaults to a unique identifier ideal for use as an application ID. Although it is recommended that you use this or a similar ID, you can choose to use any string value provided that the ID is not already used by another application listed in the ZQure manager. It is more secure to use an unrecognizable application ID for each of your applications.
- **Ensure the safety and secrecy of generated licence keys.** Licence keys generated by ZQure must be known only by the developer who issued the key and the user who registered the software. To ensure that the protection of your software is not compromised, be sure to store a safe version of a licence key when writing the key to a configuration file or database within your application. A safe version of a key cannot be used to register an instance of your software application.
- **Use the most specific form of licence key possible.** Each licence key must have a username associated with it, whereas a computer ID, an application ID and an expiry date are optional. However, together, the above elements define the scope of the licence key. The scope can be extremely wide, for example, if only a username is specified, in which case the generated key can potentially be used to register any software protected by ZQure and on any computer. The scope can be very specific, for example, if all four elements are specified, in which case the generated key can be used to register only a specific application on a specific computer and would be valid only for a certain period of time. Generally, the scope of a licence key is made as specific as possible.
- **Be sure to obfuscate.** Software written for .NET is easy to reverse engineer because .NET's MSIL is much more readable than binary machine code. Developers should be sure to use obfuscation to thwart reverse engineering of their software. Obfuscation uses a series of tricks to make MSIL much more difficult to understand without affecting the application's functionality. *Dotfuscator* is an obfuscation tool that is shipped with Visual Studio and can be used for this purpose. Please refer to the tool's documentation that accompanies Visual Studio.

6 ZQure Licences

Please refer to the following licence agreements for both the evaluation and registered versions of ZQure.

6.1 ZQure Evaluation Licence

ZQure Copyright © 2009 Sophic Edge Solutions. All rights reserved.

This license applies to the evaluation version of ZQure. If you do not have an evaluation version of ZQure, see the section titled ZQure Standard License.

Your Agreement to this Licence

By using, copying, transmitting, distributing or installing ZQure 1.0 ("ZQure"), you agree to all of the terms of this agreement ("License").

Please read the license terms below. If you do not agree to all of the terms of this License, then do not use, copy, transmit, distribute, or install this software.

Scope of Licence

This is not free software. Subject to the terms below, you are hereby licensed by Sophic Edge Solutions ("SES") to use one copy of ZQure, on one (1) computer or workstation, for evaluation purposes without charge. To access the software's full functionality, a registration fee is required. Payments must be in South African rands and be paid as described during the registration process.

Unregistered use of ZQure other than for evaluation is in violation of South African and international copyright laws.

You may, without making any payment to SES:

- a) give exact copies of the evaluation version of ZQure personally to anyone;
- b) distribute exact copies of the evaluation version of ZQure, if done exclusively through electronic channels; and
- c) make as many exact copies of this evaluation version of ZQure as you wish, for purposes of distribution as described in (a) and (b) above.

You are specifically prohibited from charging, or requesting donations, for any copies, however made, and from distributing such copies with other products of any kind, commercial or otherwise, without prior written permission from SES. SES reserves the right to revoke the above distribution rights at any time, for any or no reason.

Licence keys generated by the evaluation version of ZQure may, under no circumstances, be used on any computer other than the one they were generated on. Unauthorized use, reproduction, or distribution of this software, or any portion of it, or any licence keys generated by it is strictly prohibited.

All rights of any kind in ZQure which are not expressly granted in this License are entirely and exclusively reserved to and by SES. You may not rent, lease, modify, translate,

reverse engineer, decompile, disassemble, or create derivative works based on, ZQure, nor permit anyone else to do so. You may not make access to ZQure available to others in connection with a service bureau, application service provider, or similar business, nor permit anyone else to do so.

Warranty Disclaimers and Liability Limitations

ZQure, and any and all accompanying software, files, data and materials, are distributed and provided "AS IS" and with no warranties of any kind, whether express or implied. You acknowledge that good data processing procedure dictates that any program, including ZQure, must be thoroughly tested with non-critical data before you rely on it, and you hereby assume the entire risk of using the program. This disclaimer of warranty constitutes an essential part of this License.

Any liability of SES will be limited exclusively to refund of purchase price, if any. In addition, in no event shall SES or its principals, shareholders, officers, employees, affiliates, contractors, subsidiaries, or parent organizations, be liable for any indirect, incidental, consequential, or punitive damages whatsoever relating to the use of ZQure, or to your relationship with SES.

In addition, in no event does SES authorize you to use ZQure in applications or systems where ZQure's failure to perform can reasonably be expected to result in a significant physical injury, or in loss of life. Any such use by you is entirely at your own risk, and you agree to hold SES harmless from any and all claims or losses relating to such unauthorized use.

General

This License is the complete statement of the agreement between the parties on the subject matter, and merges and supersedes all other or prior understandings, purchase orders, agreements and arrangements. This License shall be governed by the laws of South Africa. Exclusive jurisdiction and venue for all matters relating to this License shall be in courts and for a located in South Africa, and you consent to such jurisdiction and venue. There are no third party beneficiaries of any promises, obligations or representations made by SES herein. Any waiver by SES of any violation of this License by you shall not constitute or contribute to a waiver of any other or future violation by you of the same provision, or any other provision, of this License.

Copyright © 2009 Sophic Edge Solutions. All rights reserved.

6.2 ZQure Standard Licence

ZQure Copyright © 2009 Sophic Edge Solutions. All rights reserved.

This license applies to the standard-licensed version of ZQure. If you have an evaluation version of ZQure, see the section titled ZQure Evaluation License.

Your Agreement to this Licence

By using, copying, transmitting, distributing or installing ZQure 1.0 ("ZQure"), you agree to all of the terms of this agreement ("License"). Unless you have a different licence agreement signed by Sophic Edge Solutions ("SES"), your use, distribution, or installation of ZQure indicates your acceptance of this agreement ("Licence").

Please read the license terms below. If you do not agree to all of the terms of this License, then do not use, copy, transmit, distribute, or install any copy of this software with which this Licence is included.

The terms and conditions of this License describe the permitted use and user(s) of each Licensed Copy of ZQure. For purposes of this License, if you have a valid single-copy license, you have the right to use a single "Licensed Copy" of ZQure; if you or your organization has a valid multi-user license, then you or your organization have/has the right to use up to a number of "Licensed Copies" of ZQure equal to the number of copies indicated in the documents issued by SES when granting the license.

Scope of Licence

Subject to the terms below, you are hereby licensed by Sophic Edge Solutions ("SES") to use one copy of ZQure, on one (1) computer or workstation. If ZQure is accessed through a network, a licence must be purchased for each workstation that will access it.

You may not:

- a) give exact copies of the licensed version of ZQure personally to anyone;
- b) distribute exact copies of the licensed version of ZQure; and
- c) use ZQure to implement software licencing and protection in any applications other than those developed by the person/company to which ZQure is registered.

Licence keys generated by the licensed version of ZQure may, under no circumstances, be used on any computer(s) other than the one (those) on which they were intended. Unauthorized use, reproduction, or distribution of this software, or any portion of it, or any licence keys generated by it is strictly prohibited.

All rights of any kind in ZQure which are not expressly granted in this License are entirely and exclusively reserved to and by SES. You may not rent, lease, modify, translate, reverse engineer, decompile, disassemble, or create derivative works based on, ZQure, nor permit anyone else to do so. You may not make access to ZQure available to others in connection with a service bureau, application service provider, or similar business, nor permit anyone else to do so.

Warranty Disclaimers and Liability Limitations

ZQure, and any and all accompanying software, files, data and materials, are distributed and provided "AS IS" and with no warranties of any kind, whether express or implied. You acknowledge that good data processing procedure dictates that any program, including ZQure, must be thoroughly tested with non-critical data before you rely on it, and you hereby assume the entire risk of using the program. This disclaimer of warranty constitutes an essential part of this License.

Any liability of SES will be limited exclusively to refund of purchase price, if any. In addition, in no event shall SES, or its principals, shareholders, officers, employees, affiliates, contractors, subsidiaries, or parent organizations, be liable for any indirect, incidental, consequential, or punitive damages whatsoever relating to the use of ZQure, or to your relationship with SES.

In addition, in no event does SES authorize you to use ZQure in applications or systems where ZQure's failure to perform can reasonably be expected to result in a significant physical injury, or in loss of life. Any such use by you is entirely at your own risk, and you agree to hold SES harmless from any and all claims or losses relating to such unauthorized use.

General

This License is the complete statement of the agreement between the parties on the subject matter, and merges and supersedes all other or prior understandings, purchase orders, agreements and arrangements. This License shall be governed by the laws of South Africa. Exclusive jurisdiction and venue for all matters relating to this License shall be in courts and for a located in South Africa, and you consent to such jurisdiction and venue. There are no third party beneficiaries of any promises, obligations or representations made by SES herein. Any waiver by SES of any violation of this License by you shall not constitute or contribute to a waiver of any other or future violation by you of the same provision, or any other provision, of this License.

Copyright © 2009 Sophic Edge Solutions. All rights reserved.