

## Contents

Document Revisions .....	5
Welcome .....	6
Overview .....	7
Licensing Process .....	8
Features .....	9
Prerequisites .....	11
Example Programs .....	12
Online Database .....	13
Privacy Validation .....	15
Lock Types .....	16
Custom String .....	16
Automatic .....	16
Hardware .....	16
• Hard drive serial number .....	16
• Mac address .....	16
• Processor ID .....	16
• BIOS .....	16
• Motherboard ID .....	16
• Logical Disk Serial Number .....	16
• Microsoft Windows Serial Number .....	16
• Computer Name .....	16
Lock Type Values .....	17
SoftShell Administrator Registration .....	18
Open Database .....	19
SoftShell Administrator™ .....	20
Menu Bar .....	21
Usage .....	23
Searching For Records .....	24
Application Setup .....	26

General.....	26
Option Names .....	27
User Setup.....	28
User Information.....	28
Contact Information.....	29
Registration Setup.....	30
Registration Name .....	30
Registration Setup.....	31
Application Name.....	31
User Name .....	31
Registration Will Expire.....	32
Evaluation ID .....	32
Machine ID.....	32
ID Type .....	32
Registration Number.....	32
Options.....	33
SoftShell™ Software Licensing System Architecture.....	34
SSHIClient.dll.....	35
Initialization Functions.....	36
Initialize_Simple .....	36
InitializeByParameters .....	37
InitializeByObject .....	38
SoftShell Methods.....	40
GetErrorObject.....	40
IsIDInList.....	40
IsOptionActive.....	40
CreateEvaluationIDString.....	40
DoesInternetConnectionExist .....	41
InDevelopmentEnvironment .....	41
SoftShell Properties.....	42
AppErrorDescription .....	42
AppErrorNumber .....	42

RTErrorDescription.....	42
RTErrorNumber.....	42
ErrorLocationID.....	42
ErrorMachineState.....	42
ErrorType.....	42
BIOSSerialNumber.....	42
ComputerName.....	42
DaysUntilAppExpires.....	42
EvaluationDaysRemaining.....	42
EvaluationRunsRemaining.....	43
FirstRunDate.....	43
HardDriveSerialNumber.....	43
LastRunDate.....	43
LicenseExpirationDate.....	43
LicenseMachineIDValue.....	43
LicenseOptionValue.....	43
LicenseUserName.....	43
LockType.....	43
LogicalDiskSerialNumber.....	43
MACAddress.....	43
MotherboardID.....	43
OSVersionValue.....	43
ProcessorID.....	44
WindowsSerialNumber.....	44
RegistrationStatus.....	44
Option Enabling.....	45
Encryption.....	46
SoftShell Error Values.....	47
Binary Compatibility.....	51
Deactivate Registration ID.....	52
Installation & Distribution.....	53
Files.....	53

NT Event Log .....	55
Technical Support .....	56
Security Warning.....	57
About.....	58

## Document Revisions

Version Number	Date	Reason for change
Revision 1	12/30/2009	Original – first draft/revision

## Welcome

Thank you for using the SoftShell™ Software Licensing System. We at Bevan Engineering trust you will find “SoftShell” to be secure, easy to use, and very flexible.

A large portion of Bevan Engineering business is software development. Much like you, we needed a way to allow customers to fully utilize our products during a trial period, but require the purchase of the product with continued use. We have experience with many of the tools available by software vendors.

We've tried software locks. We've tried hardware locks. We've tried the honor system. The shortcomings of all these methods gave rise to the necessity of our own licensing system. First, we developed the AppPatrol™ system – a very successful product that continues to protect many software investments. With the release of Microsoft Vista© and later - Microsoft Windows 7© - we felt it was time to take our licensing techniques to a new level. Those efforts resulted in SoftShell – a low cost and highly effective solution that works within the confines of typical user configurations and permissions. See the [features](#) section of this document to see how the SoftShell system can help you.

If you've purchased the SoftShell™ Software Licensing System, thanks very much for your business. We know you will not be disappointed. If you are evaluating SoftShell, thanks very much for giving us the opportunity to help protect your software investment.

## Overview

The SoftShell™ Software Licensing System is a set of software components designed to help protect your software investment. The SoftShell system consists of the following components:

SSHlSvc.exe - an NT Service that runs under a system account so that functions requiring administrative privileges can function normally.

SSHlClient.dll – is a 32-bit ActiveX Dynamic Link Library (.dll) that manages all licensing tasks. This .dll is the component applications using the SoftShell™ Software Licensing System will directly use.

SSHlInterface.dll - a 32-bit ActiveX Dynamic Link Library that manages requests between the SSHlClient.dll and SSHlSvc.exe components.

All of these components may be distributed royalty-free with your application. This means that once you purchase a SoftShell license, you can protect as many applications as you like with unlimited installations using the SoftShell core components.

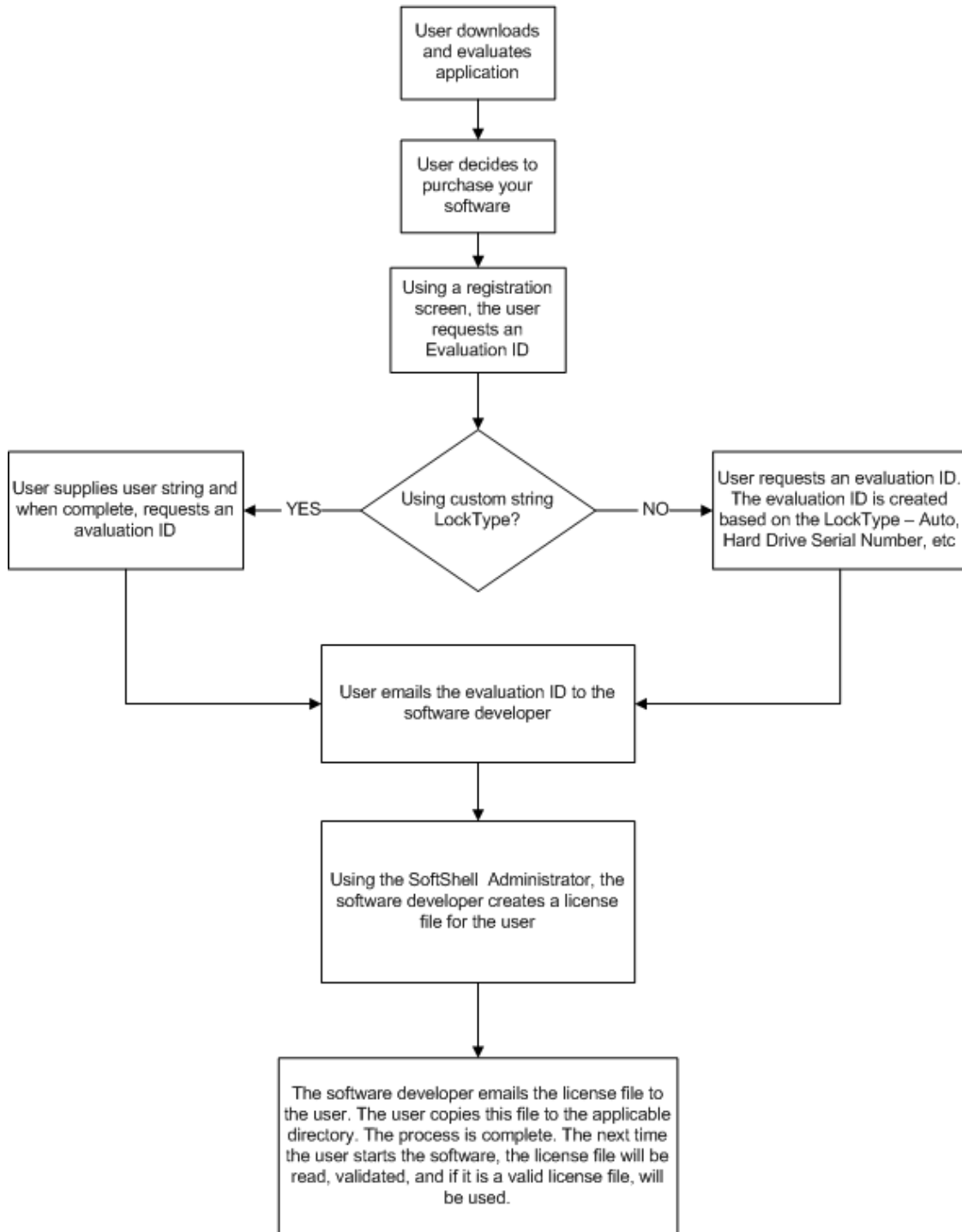
Using the SoftShell™ Software Licensing System is a fairly straightforward process. The SSHlClient.dll will return an evaluation ID based off your application's lock type (hard drive, custom string, etc). The user will email that ID to you. Using the SoftShell™ Software Licensing System Administrator, you will paste that ID into the administrator application.

The administrator will generate the license file contents needed for the requesting application. Using that information, you can create a license file that is as simple as pasting the registration ID into a text file using notepad or the text editor of your choice. You will email that license file to your user. The next time the application is started, the SSHlClient.dll will detect that file and if the file is valid, the application runs as it is registered.

If your user decides to purchase an optional feature at a later date, simply create a new license file and email it to that user. If your application is leased, you can send your user a new license file once payment is made with a new expiration date.

## Licensing Process

### Licensing Process



## Features

The SoftShell™ Software Licensing System includes the following features:

1. Does not require your application to be locked to hardware.
2. Works on all operating systems from Windows 2000 through Windows 7 in 32 and 64 bit modes (x86, x64).
3. Will run in most virtual machines and treat the virtual machine as another computer.
4. Does not require administrative access. In fact, most applications can run under a guest account while being fully protected
5. Custom security
  - Each application you develop is protected by your unique password. Nobody else - including the developers of the SoftShell™ Software Licensing System - will have access to your registration information.
6. Multiple Lock Types
  - Custom String - With the SoftShell™ Software Licensing System, you don't need to lock your application to any system identifier. Instead, you can lock the software to a custom string. This string may be a name, company, or email address. Whatever meaningful identifier serves your purpose can be used here.
  - Auto – the best type of hardware lock will be automatically found for your application based on the availability of your user's computer
  - Hard Drive – lock your software to the manufacturer's serial number. This serial number remains if the hard drive is reformatted.
  - MAC Address
  - Processor ID\*
  - BIOS Serial Number\*
  - Motherboard ID\*
  - Logical Disk – the serial number assigned to the hard drive when formatted
  - Windows Serial Number
  - Computer Name
7. Multiple Trial Types
  - Trial By Days - your application can expire after an evaluation period based on days
  - Trial By Runs - your application can expire after an evaluation period based on the amount of times it has been executed.

8. Option Enabling - you can provide your customers with a registration number that will contain information activated up to 24 premium portions of your software. If a customer purchases optional features at a later date, a new registration number can be sent (while maintaining the same lock type) that can activate those features. Any combination of optional features can be enabled.
9. Application Expiration - a registered application can be configured to expire at the date of your choosing. The number of days remaining is available for your use (warnings, etc). This is useful for "Leaseware".
10. Deactivation - Registration numbers may be stored in an internet database so that illegal copies of software can be detected and disabled.

\* May not be available on all systems – use care when selecting these lock types

## Prerequisites

The SoftShell™ Software Licensing System requires an intermediate level of programming skills. A working knowledge of Visual Basic Classic or Visual Studio.NET (VB, C#, C++) is required. A working knowledge of ActiveX DLL technology is helpful.

The sample programs included with SoftShell provide even novice programmers all the information you need to be able to license your software within minutes.

## Example Programs

Several example programs have been supplied for you in both VB6 and VB.NET 2008. The sample programs will show you the recommended way to use the SoftShell™ Software Licensing System. You may freely use and/or modify this sample code to suit your needs.

## Online Database

The SoftShell™ Software Licensing System uses an online MySQL open source database to track hardware signatures and usage data when applications are run during the evaluation period. Any computer using the SoftShell™ Software Licensing System during the evaluation period will have a hardware "fingerprint" created and this value will be stored in the database as a unique entry for your application. In the past, this was done (not necessarily by Bevan Engineering) by hiding status information in the registry, in a hidden directory, or sometimes embedded in another file (commonly called a root kit).

Two pieces of information that cannot be modified by unauthorized people must be saved while the user is evaluating your product.

1. Hardware signature – a unique hardware signature is created for each user evaluating your product. That unique identifier contains information about your hardware (typically, a hard drive serial number) and your application name.
2. Usage data – this data generally includes the day the software was first run, the last time it was run, and the number of runs (if the trial mode is by number of runs).

As you can see, no personal information that could be tied back to an individual, corporation, or any other entity is saved as it is completely unnecessary. Even so, this data is saved strongly encrypted so a person who gains "read" access to your database could find anything meaningful.

The purpose of this database is two-fold:

1. It's no secret that the key to having your application timeout after the trial period must rely on persistent data saved somewhere. Furthermore, this information must be hidden from the end user to prevent a user from resetting the trial period at will. In the past, hiding persistent data has been done using several methods (not necessarily methods used by Bevan Engineering):
  - a. Registry – persistent data is hidden somewhere in your system registry that is available to all users of the computer – usually in the HKEY\_LOCAL\_MACHINE area. This has become especially problematic because of tightened security on many computers. Administrative access must be given to your application to read and write to the registry. This may not be possible in all environments.
  - b. Hidden directories - persistent data is hidden somewhere on your computer with attributes set to hidden. Although the security settings might prevent your

application from using this method, it's not very secure. An intermediate level computer user could "unhide" files and directories while monitoring software could direct the user to the persistent data. Simply deleting the data could start the trial period over.

- c. Root kits - persistent data is hidden as a file inside another file. Sometimes, this is done using files known to be on the system regardless of whether or not your application is installed. Again, the security settings might prevent an application from using this method.
2. At the developer's request (not done automatically at start-up), the contents of the license file can be compared to license file data in the "Inactive" table. If it is found, the developer can take whatever action desired - terminating the application, displaying a warning, etc. Typically, the contents of this table will contain license data found on peer to peer networks. While generally not required for software locked to hardware, this is extremely useful for applications locked to email addresses, company names, etc. If your application is being distributed on peer to peer networks, simply adding the contents of that license file to the Inactive table would render that license file useless on all computers using it.

The structure of the database is as follows:

"Active" table:

- ID = the unique hardware identifier created for the PC evaluating your software. This encrypted data includes the application name and a mixture of hardware values.
- Value = an encrypted string that contains a date/time stamp of when the application was run, number of runs, etc.

"Inactive" table:

- ID = the encrypted registration string created by the SoftShell administrator.

By storing this hardware information, several problems are resolved:

1. No hidden objects are created on a user's machine
2. No special permissions are required to read and write registry information
3. No root kits are required

As a developer using the SoftShell™ Software Licensing System, you have two options in terms of how you want this data stored during your application's evaluation period:

1. The Bevan Engineering database - users evaluating your application can use the Bevan Engineering online database. Since there is no personal information stored, the data is stored as an encrypted text string using your password (even Bevan Engineering cannot decrypt your data), and access to that database is strongly protected, there is no risk in using this database. To use this database, you do nothing - the SoftShell™ Software Licensing System defaults to the Bevan Engineering database. You are free to use this database as often as you like.
2. Create your own database - download the database from the Bevan Engineering support page and install on your server system. If your domain is hosted by a third party, check with your technical support for MySQL database support. Most provide MySQL capabilities for free, but it may need to be activated. Once you install the database on your server, simply "point" your application to this database by providing the login information to the SoftShell client as optional parameters. Once the client detects you are redirecting the database source, the default values are not used and your database will be used instead.

Because of the fear some users may have in storing hardware information to a server - a fear that is unfounded (unless someone has a method to tie (for example) a processor ID back to the person using it - an impossible task in our opinion - it is recommended you make a privacy statement in your End User License Agreement (EULA) when someone installs your software for evaluation. Any objections to storing this data can be handled during this phase of evaluation.

**Privacy Validation** – if there continues to be concerns regarding the information stored in the online database, a method exists to verify no personal information is saved. Using an example program – or your program – set the password as an empty string (e.g. `n_strPassword = ""`, `n_strPassword = String.Empty`) when calling one of the initialization methods. This will cause the encryption/decryption functions to be skipped. Using your MySQL administrator, simply view the newly created record(s) in plain text.

Note that an empty password string is not a valid value. Thus, your application will not be able to be setup in the administrator application and no registration ID required for the license file can be created. This feature is provided solely for validation purposes.

## Lock Types

The SoftShell™ Software Licensing System offers several methods to "lock" your software to a user. They include:

**Custom String** - With the SoftShell system, you don't need to lock your application to hardware or software. Instead, you can lock the software to a custom string. This string may be a name, company, or email address. Whatever meaningful identifier serves your purpose can be used here. This is useful for site licenses, multi-computer usage, etc.

**Automatic** – Selecting the automatic lock type is the recommended method of locking your application to hardware. The SoftShell system will lock to the best component or device it can find. While usually the hard drive serial number, it will find the best component it can find if the hard drive serial number cannot be resolved

**Hardware** – Realizing you know your users and customers better than anybody, the SoftShell system will allow you to lock to the specific device or component you desire. Take care in using this option as some systems or configurations may not support your hardware choice. But if you choose to lock to a specific device, you have a wide choice of components and devices:

- **Hard drive serial number** - This is the serial number created by the manufacturer – not the operating system
- **Mac address**
- **Processor ID**
- **BIOS**
- **Motherboard ID**
- **Logical Disk Serial Number** - This is the serial number created by operating system when the drive was formatted
- **Microsoft Windows Serial Number**
- **Computer Name**

## Lock Type Values

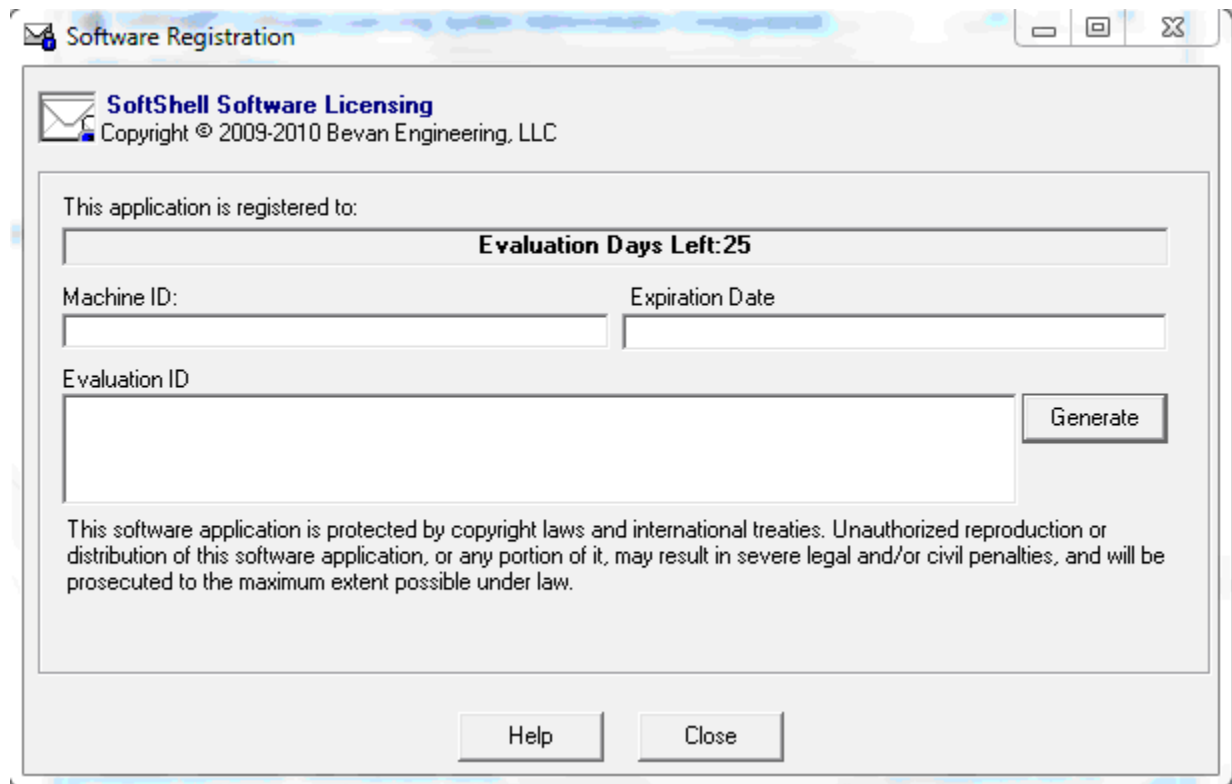
The following Lock Type values are used. The data type called "RegistrationLockType" will return a value of one of the following:

- 1** - (LOCK\_TO\_AUTO): The SoftShell system will lock to the best component or device it can find. While usually the hard drive serial number, it will find the best component it can find if the hard drive serial number cannot be resolved.
- 2** - (LOCK\_TO\_STRING): Lock to any string value (email address, company, etc).
- 3** - (LOCK\_TO\_HD\_SERIAL): Lock to a hard drive serial number
- 4** - (LOCK\_TO\_MAC\_ADDRESS): Lock to the user network card MAC address
- 5** - (LOCK\_TO\_PROCESSOR\_ID): Lock to the processor ID
- 6** - (LOCK\_TO\_BIOS) : Lock to the BIOS ID/Serial number
- 7** - (LOCK\_TO\_MOTHERBOARD\_ID): Lock to the processor ID
- 8** - (LOCK\_TO\_LOGICAL\_DISK): Lock to the first logical disk serial number
- 9** - (LOCK\_TO\_WIN\_SERIAL) : Lock to the Microsoft Windows Serial number
- 10** - (LOCK\_TO\_COMPUTER\_NAME) : Lock to the user's computer name

## SoftShell Administrator Registration

You are permitted to use the full functionality of the SoftShell Administrator™ for a period of 30 days at no cost to you. Before that time period expires, you must purchase a license for continued use of the application. The steps required to register the application are as follows:

1. Using the purchasing option of your choice, purchase a license of the SoftShell Administrator™.
2. Click on the "Generate" button. An evaluation ID will be created for you.
3. Copy that evaluation ID and email it to [registration@bevanengineering.com](mailto:registration@bevanengineering.com)



When you have received your license file in email, simply copy that file to the SoftShell Administrator™ installation directory. Your application will be registered.

## **Open Database**

Before the SoftShell Administrator™ can be used, a database must be opened. Using the Main Menu, select File->Open. A dialog will be displayed allowing you to select any SoftShell database (in the event you have multiple copies). Once this selection has been made and the database has been opened, this information will be saved to the system for continued use.

## SoftShell Administrator™

Quite simply, the SoftShell Administrator™ will generate the correct license file contents needed for a user to register your product. After setting the required parameters, a license file can be created and sent to your customer.











The SoftShell Administrator™ can also manage all your application registration needs. User settings, application settings, and registration settings are saved in a Microsoft Access© database for you. You are free to make multiple copies of this database and select which one you would like to use at any time. If you prefer multiple databases, feel free to download as many copies as you require from the Bevan Engineering [support web page](#).

## Menu Bar

A menu bar is used for most configuration screens to navigate through records in database tables.



Although a database connection has been established, no records become available until you choose which records you wish to access. Typically, "Get All" will be used and all records related to the current setup screen will be available. The menu bar will help navigate through these records.

Button	Name	Purpose
	New	This will create a new record.
	Save	This will save all current screen entries
	Delete	This will delete the current record. Use extreme caution as this information may not be recoverable.
	Get All	This will retrieve all records related to the current setup screen
	Search	This button will display a search form. Only records specified by your search query will be returned.
	First	The first record of the currently loaded recordset will be displayed.
	Back	The previous record of the currently loaded recordset will be displayed.
	Next	The next record of the currently loaded recordset will be displayed.
	Last	The last record of the currently loaded recordset will be displayed.
	Help	The help system will be displayed.



Close

The current screen will be closed.

## Usage

When the “Get All” menu button is clicked, all records from the table the screen controls (for example, the “Applications Setup” screen is used to add, edit, or delete records in the “Applications” table in the Microsoft Access database you are connected to) are loaded into a recordset. That recordset is traversed using the navigation controls (first, last, next, previous).

If the “Search” button were clicked, you would enter the values to search on. Only records meeting the criteria of your search would be loaded into a recordset that again, could be traversed using the navigation controls. More information can be found in the [Searching For Records](#) section of this document.

When the recordset is initially loaded, the first record in the recordset will be displayed. That record can either be edited or deleted.

If a new record is required, click the “New” button. Once all information is entered into the screen, the “Save” button can be clicked to insert that new record into the recordset.

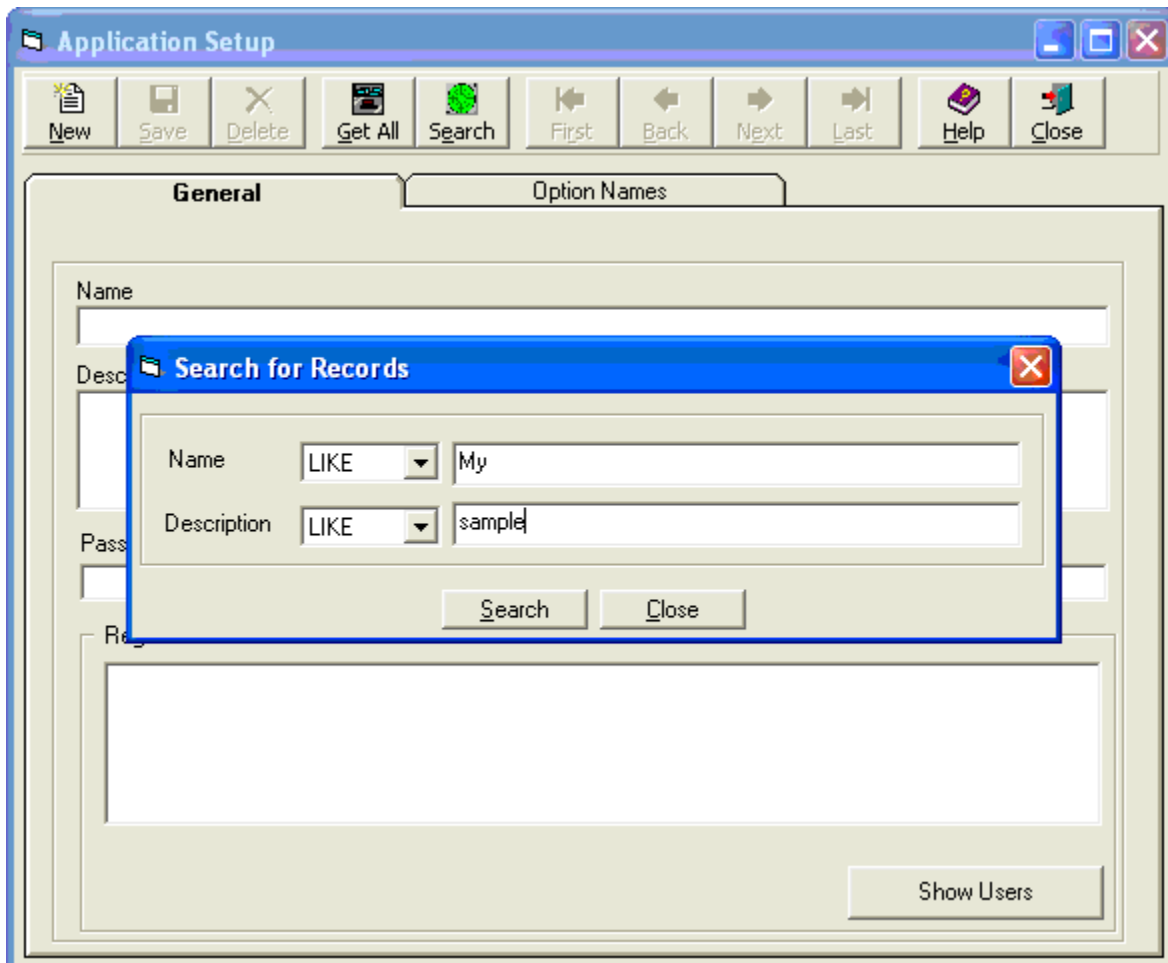
If a specific record is required, some screens may allow the “Find” button to be used. Clicking on the “Find” button will display a dialog that will allow you to select/enter your search criteria. When complete, the first record – if one exists – will be displayed. Clicking the “Find Again” button will advance to the next record in the recordset matching your criteria (obviously, if one exists).

To summarize, the “Get All” and “Search” buttons will load a set of records into a recordset. All other buttons will allow you to traverse through that recordset as desired.

## Searching For Records

Some setup screens allow the searching of related records. For example, if the "Get All" button were clicked in the Applications Setup Screen, all application records would be returned. From that point on, the menu bar navigation buttons (first, previous, next, last, etc) would traverse all application records in the database. However, if the amount of records become too great, this method may be too cumbersome. To alleviate this potential problem, the search screen can be used to limit the amount of records to a more manageable size.

When you click on the "Search" menu button, a screen will be displayed similar to this:



Records can be searched by either name, description, or if in the user screen, by company. Any combination of name and description/company can be used.

In this example, the "Name" dropdown list contains the search qualifiers. The text box to its right accepts the data you wish to search for. So, the above sample would search for all application records that contain the word "My". The LIKE qualifier means that if a name in the application table contains what has been input to the text box, it will be returned in the application setup screen. You can narrow this search by using the many qualifiers also available to you (i.e. >,<,<>).

The "Description" (and in the case of Users, "Company") dropdown list does the same thing as described above in the "Name" dropdown list. Instead, this query will return all records based on the description/company.

Either search query can be used. Additionally, both can be used. The search feature is not very robust, but should be adequate to narrow the amount of records you wish to deal with significantly.

## Application Setup

When a new application is available for purchase, the unique application must be setup in the database. Use the Menu Bar to add, edit, delete, or navigate through all application records.

**General** - The name, description of the application, and the password used with the SoftShell code in your application is required. The "Registered Users" list will show the users that have registered this software (for display purposes only).

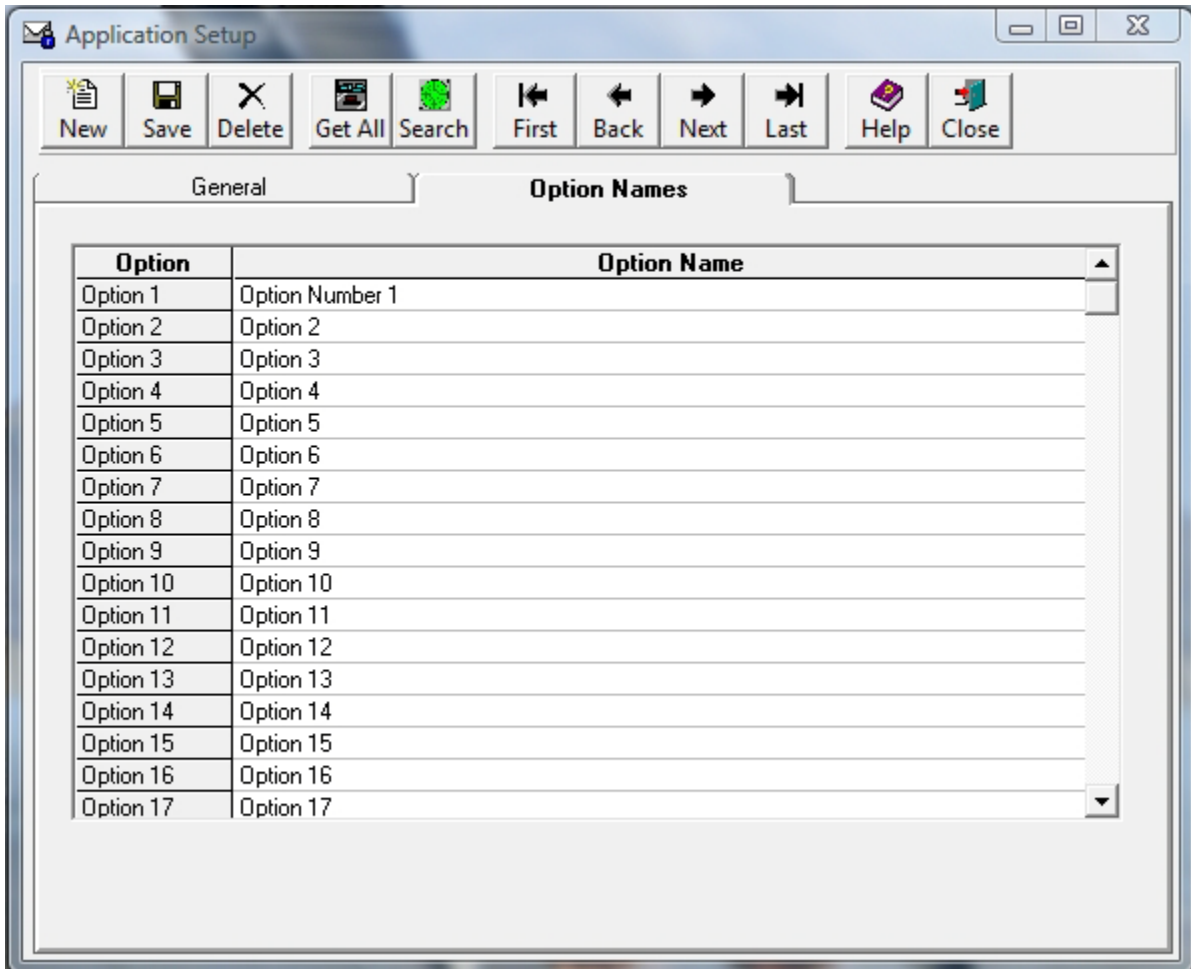
**The password must match *exactly* what you use in your software application when initializing the SoftShell system.**

The screenshot shows a window titled "Application Setup" with a menu bar containing: New, Save, Delete, Get All, Search, First, Back, Next, Last, Help, and Close. The "General" tab is selected, and the "Option Names" section is visible. The form contains the following fields and data:

- Name:** SoftShellAdmin
- Description:** Soft Shell Licensing System Administrator
- Password:** mypassword
- Registered Users:**
  - TestUser10 Registration
  - TestUser11 Registration
  - TestUser12 Registration
  - TestUser13 Registration
  - TestUser14 Registration

A "Show Users" button is located at the bottom right of the form.

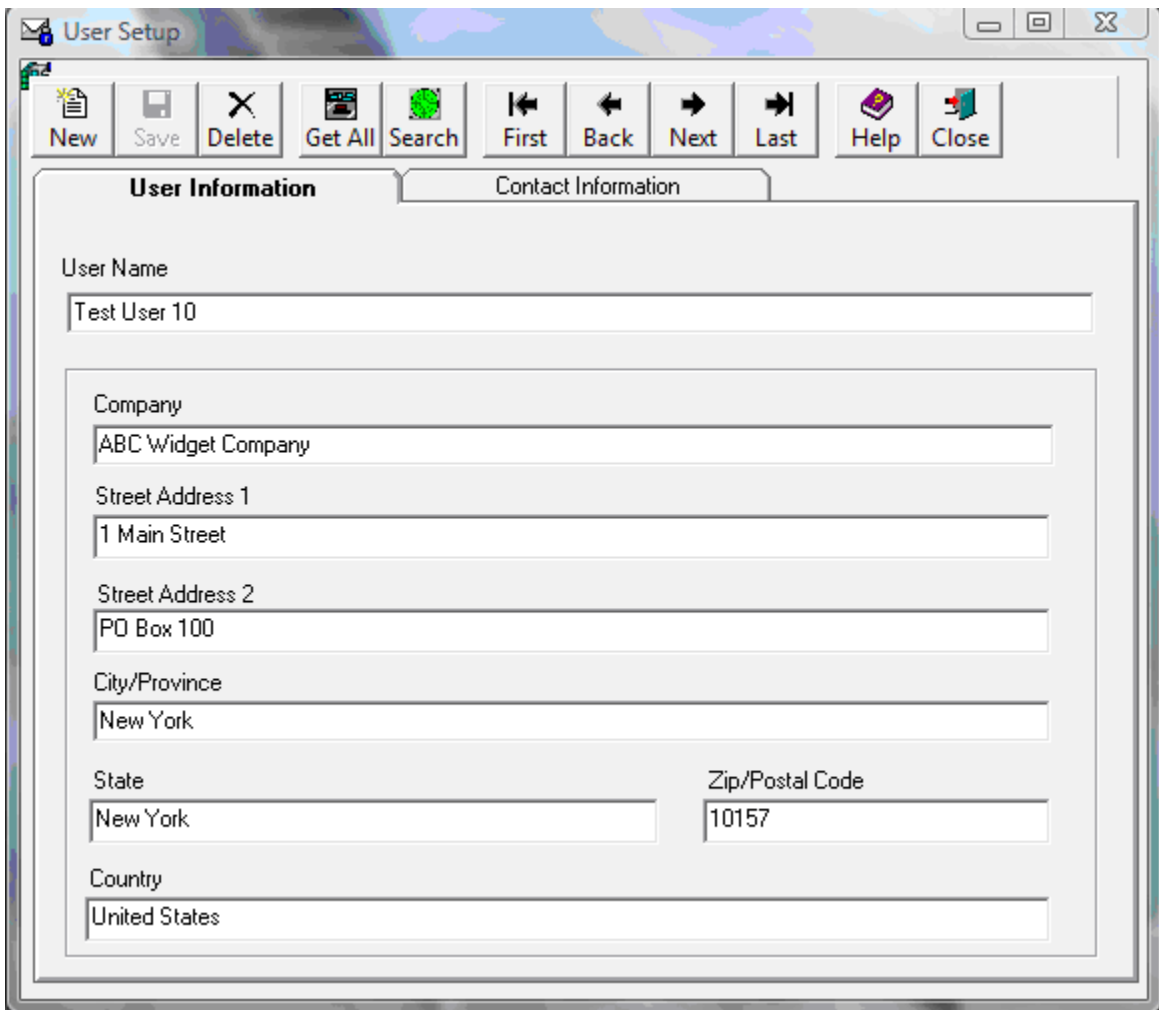
**Option Names** - Up to 24 options can be configured for each software application. Double-Click on a cell under "Option Name" to edit the name of the option to something meaningful.



## User Setup

When a customer purchases your software, the unique user must be setup in the database. Use the Menu Bar to add, edit, delete, or navigate through all user records.

**User Information** - The user name is required. All other information is optional.



The screenshot shows a window titled "User Setup" with a menu bar containing the following items: New, Save, Delete, Get All, Search, First, Back, Next, Last, Help, and Close. Below the menu bar are two tabs: "User Information" (selected) and "Contact Information". The "User Information" tab contains the following fields:

User Name	
Test User 10	
Company	
ABC Widget Company	
Street Address 1	
1 Main Street	
Street Address 2	
PO Box 100	
City/Province	
New York	
State	Zip/Postal Code
New York	10157
Country	
United States	

**Contact Information** - All contact information is optional, but it is highly recommended you save the user's email address and at least one phone number.

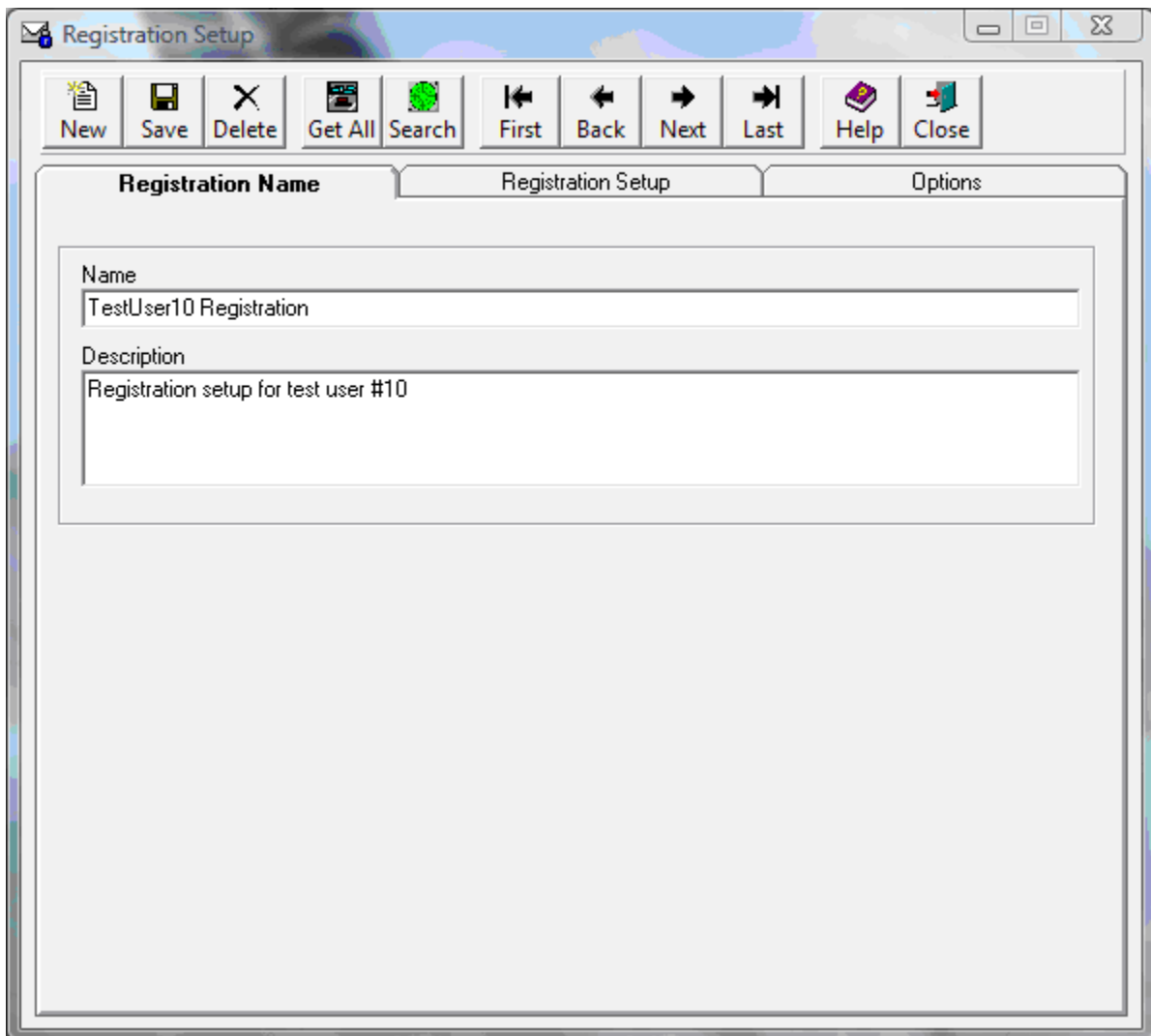
The screenshot shows a 'User Setup' dialog box with a 'Contact Information' tab selected. The dialog has a menu bar with 'New', 'Save', 'Delete', 'Get All', 'Search', 'First', 'Back', 'Next', 'Last', 'Help', and 'Close'. The 'Contact Information' section contains several text input fields:

E Mail Address	
<input type="text" value="testuser@widgetsrus.com"/>	
Web Site	
<input type="text" value="www.widgetsrus.com"/>	
Office Phone	Home Phone
<input type="text" value="212-555-1212"/>	<input type="text" value="212-555-0000"/>
Cell Phone	Pager
<input type="text" value="212-555-2121"/>	<input type="text" value="212-555-9999"/>
Fax Number	
<input type="text" value="212-555-1122"/>	

## Registration Setup

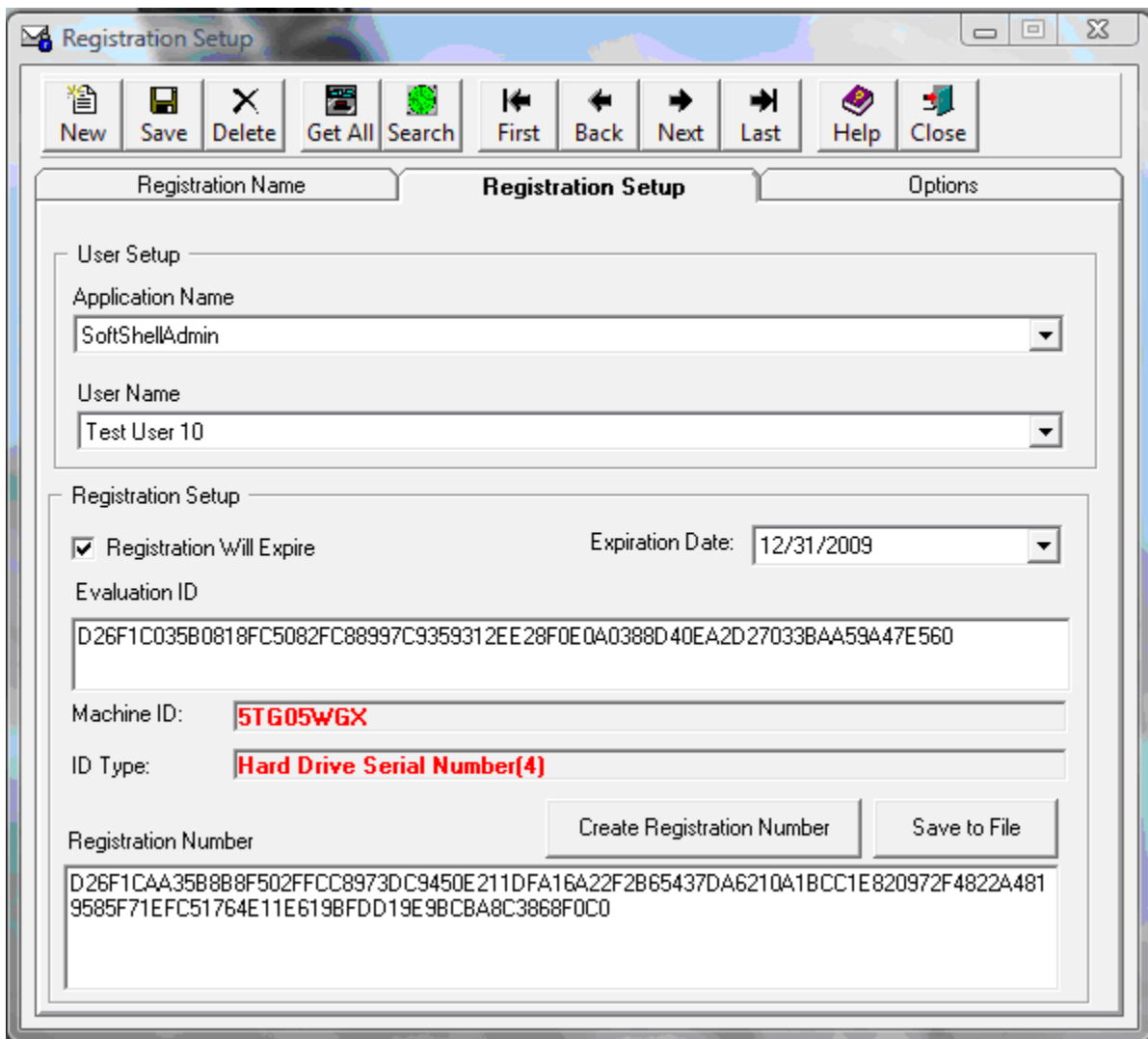
When a customer purchases your software, the unique registration information must be setup in the database. Use the Menu Bar to add, edit, delete, or navigate through all registration records.

**Registration Name** - A unique registration name must be entered. This name could be an order number, a date/time stamp, or any type of identifier meaningful to you. The description field is optional.



The screenshot shows a window titled "Registration Setup" with a menu bar and a main form area. The menu bar contains the following items: New, Save, Delete, Get All, Search, First, Back, Next, Last, Help, and Close. The main form area is divided into three sections: "Registration Name", "Registration Setup", and "Options". The "Registration Name" section contains two text input fields: "Name" with the value "TestUser10 Registration" and "Description" with the value "Registration setup for test user #10".

**Registration Setup** - All information required to create a registration number is entered into this tab of the Registration Setup screen.



**Application Name:** A dropdown list of all applications currently available for registration.

**User Name:** A dropdown list of all previously configured users. The user required for the current registration must have been previously configured.

**Registration Will Expire:** If checked, the registration number will only be valid through the date in the "Expiration Date" box.

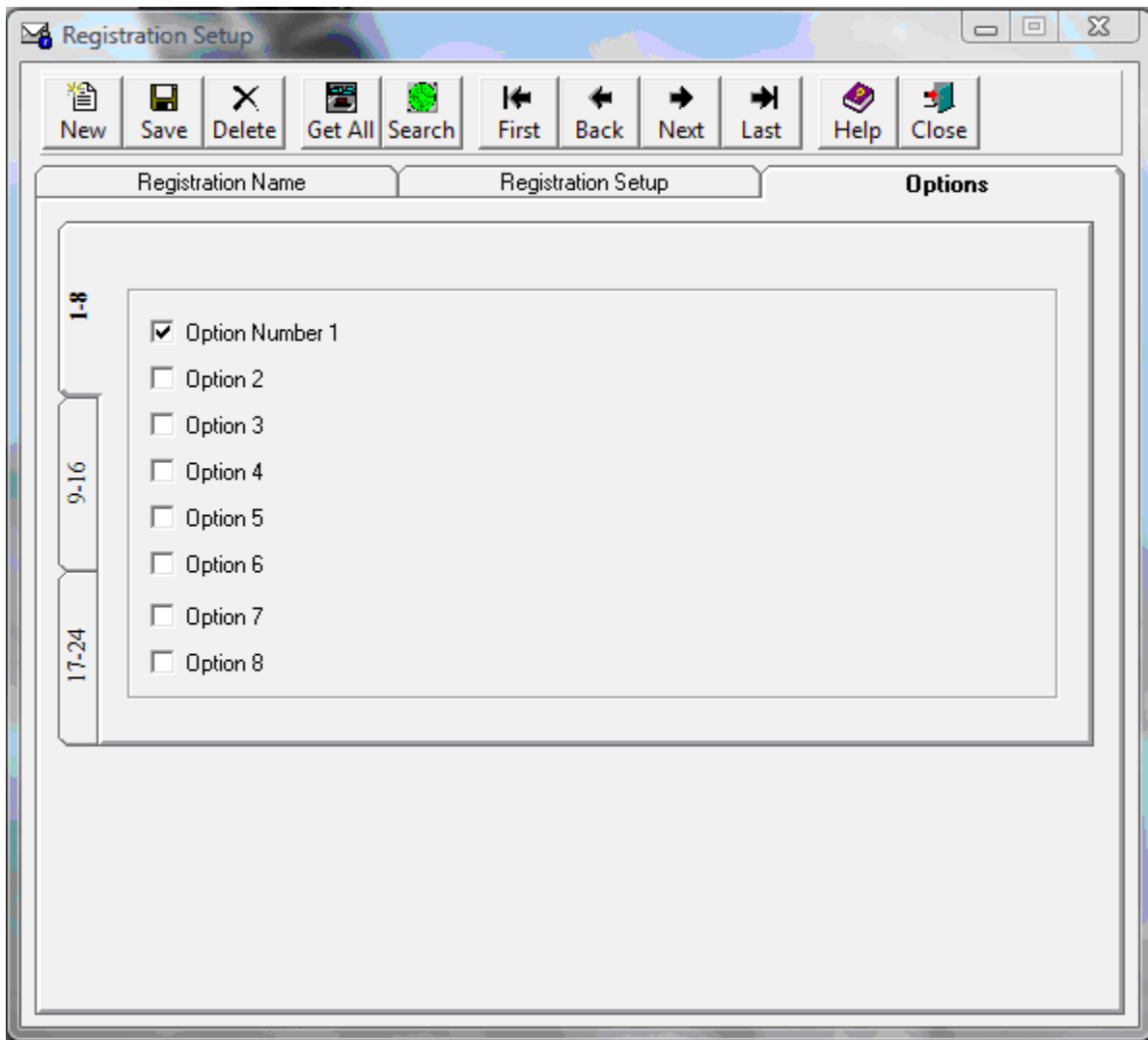
**Evaluation ID:** The user must send you an evaluation number generated on their computer. If you've specified a custom string LockType, your registration screen should include a text box that allows the user to type that value in (feel free to use/modify the registration screen included with the sample programs). When the user clicks on a button that calls the **CreateEvaluationIDString** method, an encrypted string will be built and returned as a string from that public method using that value. Otherwise, an encrypted string will be built and returned as a string from that public method using whatever LockType you require (auto, hard drive, MAC address, etc)

**Machine ID:** This field will display the decrypted value used to identify the machine. If the LockType is a custom string (email address, company name, etc), that string should be displayed

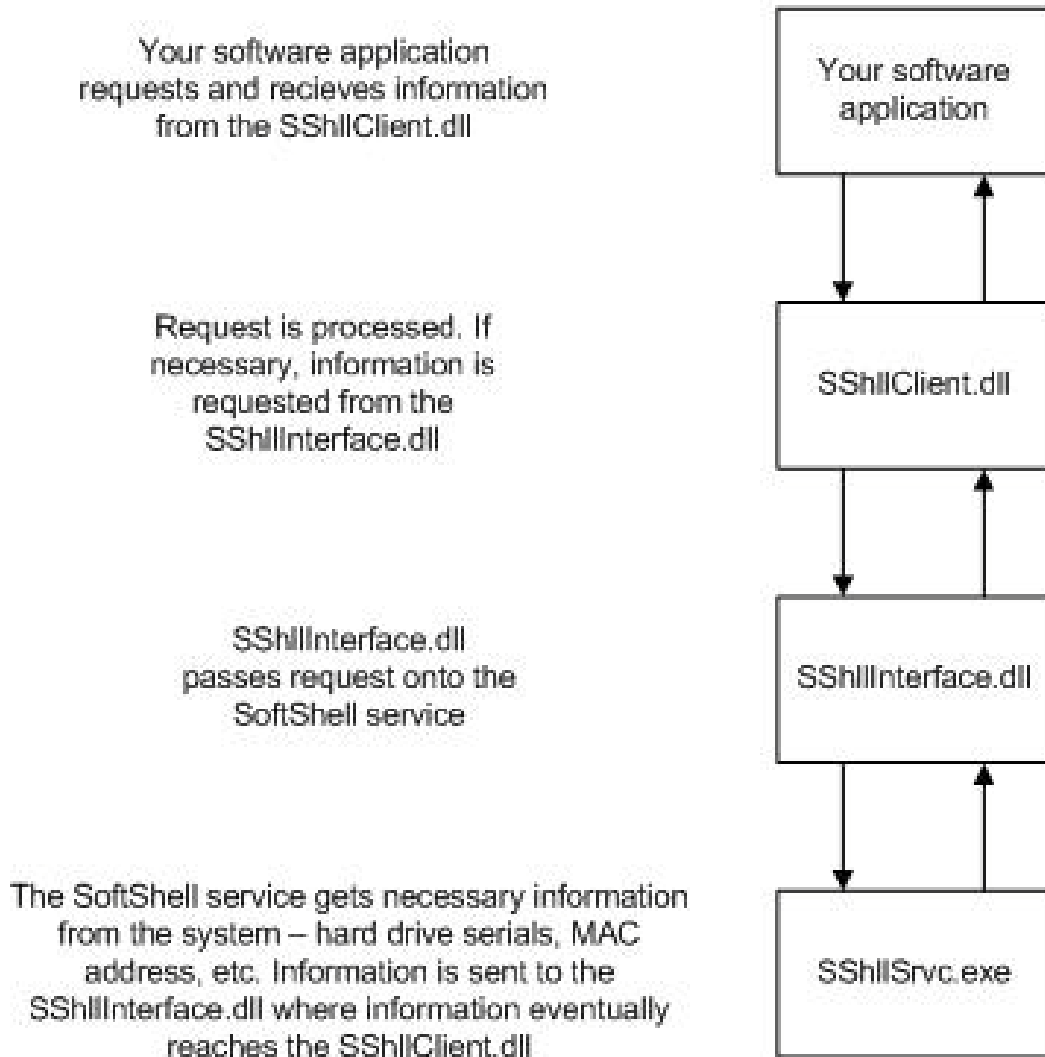
**ID Type:** This displays the type of lock when the evaluation ID was created. The number in parentheses is an internal code used for diagnostic purposes by Bevan Engineering.

**Registration Number:** If all information is entered, clicking the "Create Registration Number" button will generate a registration ID for the current configuration. This ID is used to create a license file. By clicking on the "Save To File" button, the registration ID will be saved to a file and location of your choice. That file can then be sent to your user. When the user copies that file to the directory you specify, that application will be licensed.

**Options:** Click on an option to "activate" the desired feature for the current user. A new registration ID must be generated if these settings are modified after the last registration ID creation.



## SoftShell Licensing System Architecture



## SShllClient.dll

The SShllClient.dll component is the primary component of the SoftShell™ Software Licensing System and can be widely distributed. This DLL controls all aspects of your user licensing process and protects your software. This DLL is the link between your application and the interface DLL – the component that communicates directly with the SoftShell Service Application

Only one SShllClient.dll component should exist in the System32 directory. One .dll can manage dozens of applications on one computer. The amount of applications protected by the SShllClient.dll are unlimited and the amount of applications protected on one system is unlimited.

Unless specified, all versions of the SShllClient.dll are compatible. Of particular importance, binary compatibility must be maintained. If a new version of SShllClient.dll becomes available that specifically states binary compatibility has been broken, a new version of your software with the new SShllClient.dll must be compiled and distributed.

More information about binary compatibility can be found [here](#).

## Initialization Functions

Before your application can begin calling methods and accessing property values, the SoftShell client (SSHllClient.dll) must be initialized. There are several different methods available.

### Initialize\_Simple

This initialization method is the simplest and used the most. The only parameters required are:

1. Application name – the unique name of your application
2. Password – your unique password. This password is used for encryption purposes. Text will be encrypted and decrypted using this value. For your own protection:
  - a. Make sure your password is at least 8 characters long, uses a mixture of upper and lower case letters, uses numeric values, and contains special characters (%,#,&, etc)
  - b. Do not use literal text in your application. A hex editor may be able to display your password. Instead, build your password string using the Chr command or whatever comparable command your programming environment uses. For example, in VB6 we would build the string for “pass” as follows:

```
Dim n_strPassword as string
```

```
n_strPassword = Chr(112) & Chr(97) & Chr(115) & Chr(115)
```

- c. As noted in the [Privacy Validation](#) section of this document, setting this password to an empty string will cause evaluation values to be saved as unencrypted text. An empty string is not a valid value, but this feature is provided for verification purposes only.
3. License file path - the path your license file can be found. Typically, that will be in the VB6 “App.Path” property or in VB.NET, the “Application.Info.DirectoryPath” property.
4. TrialType (optional) – the type of trial available for your application. The default value is 0 - trial by days.:

0 = trial by days – the number of days your application will run during the evaluation period before expiring

1 = trial by runs - the number of times your application will run during the evaluation period before expiring

5. TrialPeriod – the number of days or the number of time your application can be run during the evaluation period. The default value is 30.

### InitializeByParameters

This initialization method offers the most control. All parameters except for database login values are required:

1. Application name – the unique name of your application
2. Password – your unique password.
3. License file path - the path your license file can be found.
4. TrialType – the type of trial available for your application. Available values are:

**0** = trial by days – the number of days your application will run during the evaluation period before expiring

**1** = trial by runs - the number of times your application will run during the evaluation period before expiring

5. TrialPeriod – the number of days or the number of times your application can be run during the evaluation period.

6. LockType – the type of lock you prefer:

**1** = Lock to Auto – the SoftShell system will automatically find the best hardware lock for you. This is the recommended value because not all systems may have the hardware lock availability you prefer. For example, you may wish to “lock” your license/registration to a BIOS serial number. Some computer manufacturers may not make that number available. If it is not available, the SoftShell initialization method will fail.

**2** = String – instead of locking to hardware, your software can lock to a custom string such as an email address, company name, or user name.

**3** = Hard drive serial number – the manufacturer’s serial number will be used for licensing/registration. This serial number cannot be changed and is the first type of hardware lock the SoftShell system will use if a value of one (1) – lock to auto – is used.

**4** = MAC Address – lock to the first Ethernet enabled NIC card MAC address.

**5** = Processor ID – lock to the first processor ID found \*

**6** = BIOS serial – lock to the BIOS serial number \*

**7** = Motherboard ID – lock to the motherboard serial number \*

**8** = Logical disk – lock to the serial number assigned by the OS. Use care in requesting this lock type because the next time your users format their hard drive, they will likely need a new registration ID/license file.

**9** = Windows serial number – lock to the OS serial number. Use care in requesting this lock type as this value will become obsolete if the user replaces the operating system – as opposed to an upgrade.

**10** = Computer Name – locks to the computer name. Use care as computer names can be changed very easily.

7. Allow Evaluation Without Internet Connection - . Available values are:

**False** – an internet connection must exist if a user wants to evaluate your software product. The internet connection is required to utilize the online database.

**True** – if an internet connection does not exist, all trial information will be skipped and your application will be allowed to run in evaluation mode

8. License file path - the path your license file can be found.

9. License file extension (optional – the SoftShell system will look for the first “.lic” file it finds in your license file path by default. However, by changing this string value to another extension (e.g. “txt”), the system will look for the first file matching your extension for licensing information.

10. Database server host (optional) – the host of your online database

11. Database port number (optional) – the port number of your online database

12. Database user name (optional) – the user name required to login to your database

13. Database password (optional) – the password required to login to your database

14. Database name (optional) – the database name attached to your server

\*May not be available on all computer systems

## InitializeByObject

This initialization method offers accepts a data object as a parameter. Instead of using the InitializeByParameters, a class object can be configured. That class object can be passed into the InitializeByParameters method.

For help using this method, download the InitByObject VB6 example.

These initialization methods are perhaps the most important functions called in your application code. They "set up" the ActiveX DLL to be aware of your application, password, trial period, etc. No other method can be called and no property can be referenced before this method is called or unexpected behavior will result.

In looking at the parameters passed into these functions, you will notice that one of the parameters is your unique password for your application. Almost all functions contained in the SShellClient.dll rely on your unique password. Even the developers of the SoftShell™ Software Licensing System do not have access to your licensing

parameters. Be sure to keep this password in a safe place and assuming you store your important data in the SoftShell Microsoft Access© database file, make backups frequently. **Bevan Engineering cannot recover lost passwords.**

## SoftShell Methods

The following SoftShell client (SShllClient.dll) methods can be called once an initialization function has been called and the .dll is initialized:

**GetErrorObject** (VB6 Only) – By passing in an error object class to this method, the data object will be updated with all error values. The error object class can be downloaded from the Bevan Engineering support web page.

Returns - A boolean value. A value of true means the function succeeded without error.

Parameter #1 - ErrorObject

- Type: clsErrorObject

**IsIDInList** - Checks to see if the registration ID located inside the license file is in the Inactive table.

Returns - A boolean value. A value of true means the registration ID is in the Inactive table.

**IsOptionActive** - Checks to see if the option value passed in as a parameter has been activated in the license file.

Returns - A boolean value. A value of true means the option has been activated.

Parameter #1 - OptionValue

- Type: Long
- Description: The value number of an optional feature of your software.

**CreateEvaluationIDString** – Creates an evaluation string based off either the LockType or user string. If no string is passed in as a parameter, an evaluation string will be created using the LockType. Otherwise, the evaluation string will be created using the string passed in as a parameter.

Returns - A string value – the evaluation string. A zero length string means an error occurred.

Parameter #1 (optional) – UserString

- Type: String
- Description: A custom user string – email address, company name, user name, etc

**DoesInternetConnectionExist** - Checks to see if the user's computer is connected to the internet

Returns - A boolean value. A value of true means an internet connection exists.

**InDevelopmentEnvironment** - Checks to see if the user is in a development environment such as Microsoft Visual Studio© or Borland Builder©

Returns - A boolean value. A value of true means a development environment was found to be running.

## SoftShell Properties

The following SShellClient.dll properties are exposed for public use:

**AppErrorDescription** (String) - A textual description of an application error that occurred. "Application Errors" are defined as errors that are generated by the application e.g. supplying an invalid parameter, failing to provide information, missing data, validation errors, etc. Only English is supported at this time.

**AppErrorNumber** (Long) - The error number of the application error that occurred. The list of application error numbers can be found [here](#). You can replace the default English text with verbiage of your choice by referencing this error number.

**RTErrrorDescription** (String) - A textual description of a runtime error that occurred. "Runtime Errors" are defined as errors that are generated by the operating system e.g. invalid cast, divide by zero, null exception, etc. Only English is supported at this time.

**RTErrrorNumber** (Long) - The error number of the runtime error that occurred. The list of runtime error numbers can be found [here](#). You can replace the default English text with verbiage of your choice by referencing this error number.

**ErrorLocationID** (Integer) - an enumerated value pointing to the SoftShell function that failed.

**ErrorMachineState** (Integer) - The logic for the SoftShell system follows state machine logic. This value shows the state or "step for diagnostic purposes.

**ErrorType** (Integer) - The error type could be an application error (invalid entry, missing data, etc) or a system error (error opening database, etc). This value indicates which type of error has occurred in the event you wish to handle either differently.

**BIOSerialNumber** (String) - The BOIS serial number - may not be available on all systems.

**ComputerName** (String) - The name of the user's computer

**DaysUntilAppExpires** (Long) - The number of days remaining if an application is set to expire after a certain date (leaseware). Note that this property is **not** related to the evaluation period. That property is *EvaluationDaysRemaining*.

**EvaluationDaysRemaining** (Long) - The number of days remaining in the evaluation period.

**EvaluationRunsRemaining** (Long) - The number of runs remaining in the evaluation period

**FirstRunDate** (String) – The date the application was first run.

**HardDriveSerialNumber** (String) - The hard drive serial number of the user system

**LastRunDate** (String) – The last date the application was run.

**LicenseExpirationDate** (String) – The date the application will expire. A zero length string will be returned if no expiration date exists.

**LicenseMachineIDValue** (String) – The machine ID found in the license file

**LicenseOptionValue** (long) – The option value found in the license file

**LicenseUserName** (String) – The name of the user found in the license file

**LockType** (Registration\_LockType) – Get the LockType currently set. Enumeration can be found [here](#).

**LogicalDiskSerialNumber** (String) – The serial number of the first hard disk found as assigned by the operating system

**MACAddress** (String) – The MAC address of the first NIC card connected to the network/router

**MotherboardID** (String) – The motherboard ID as assigned by the manufacturer. *May not be available on all systems.*

**OSVersionValue** (integer) – An enumerated value indicating the operating system:

**1** = Windows 95

**2** = Windows 98

**3** = Windows ME

**4** = Windows NT version 3.51

**5** = Windows NT version 4.00

**6** = Windows 2000

**7** = Windows XP

**8** = Windows Server 2003

**9** = Windows Vista

**10** = Windows 7

**ProcessorID** (String) – The processor ID as assigned by the manufacturer. *May not be available on all systems.*

**WindowsSerialNumber** (String) – The Microsoft Windows© operating system serial number.

**RegistrationStatus** (Registration\_Status) – Gets the registration status. The available registration values are:

**-99** - (APP\_REG\_ERROR): An error occurred attempting to initialize the SShlClient.dll. The reason for an error can be found in the error properties.

**0** - (APP\_IS\_REGISTERED): The application is registered

**1** - (APP\_IN\_TRIAL): The application is being executed during the evaluation period.

**2** - (TRIAL\_IS\_EXPIRED): The evaluation period has expired

**3** - (LICENSE\_IS\_EXPIRED): The license has expired

**99** - (APP\_NEVER\_USED): The application has been executed the first time. May be useful for welcome screens, licensing screens, etc

## Option Enabling

Up to 24 optional features can be activated/deactivated via the registration ID in the license file. By using the `IsOptionActive` method (see `SoftShellUtils.dll` Methods), your application can determine whether an optional feature has been acquired by your user. If your user decides to acquire an optional feature at a later date, simply create a new license file with the option enabled (via a checkbox on the registration setup screen) and email the new license file to your user. Once the new license file has been copied into the appropriate directory, the `IsOptionActive` method (of course, with the feature number passed in as a parameter) will return `TRUE`.

## Encryption

SoftShell uses the latest in encryption technology to help protect your software. The type of encryption and the way it is managed is proprietary information. No information will be provided regarding this issue at any time.

The encryption used relies on a password supplied by you. Nobody - including Bevan Engineering - can "crack" your encrypted data to gain access to your registration details.

## SoftShell Error Values

If the ErrorType property = 1 (APP\_ERROR), the following error numbers become available to your application.

**100** APP\_ERROR\_MISSING\_DATES: Evaluation data was retrieved, but one or more dates are invalid or missing

**101** APP\_ERROR\_CLOCK\_SET\_BACK: The system clock has been set back.

**102** APP\_ERROR\_INVALID\_APP\_NAME: The application name cannot be zero-length

**103** APP\_ERROR\_INVALID\_PASSWORD: The password cannot be zero-length

**104** APP\_ERROR\_INVALID\_TRIAL\_PERIOD: The trial period must be a numeric value  $\geq 0$

**105** APP\_ERROR\_INVALID\_LOCK\_TYPE: The lock type must be a numeric value  $\geq 1$  and  $\leq 10$

**106** APP\_ERROR\_COMPUTER\_NAME\_NOT\_DETECTED: The computer name could not be determined

**107** Unused

**108** Unused

**109** APP\_ERROR\_NOT\_INITIALIZED: An operation was attempted before initializing the SShlClient.dll with one of the initialization methods

**110** Unused

**111** Unused

**112** APP\_ERROR\_INVALID\_USER\_STRING: The user string cannot be zero-length when using LockType LockToString

**113** APP\_ERROR\_OPTION\_VALUE\_OUTSIDE\_LIMITS: The option value must be a numeric value  $\geq 1$  and  $\leq 24$

**114** APP\_ERROR\_AQUIRING\_PROVIDER: Encryption error

**115** APP\_ERROR\_BAD\_HASH\_TYPE: Encryption error

**116** APP\_ERROR\_ERR\_BAD\_ENCRYPTION\_TYPE: Encryption error

- 117** APP\_ERROR\_CREATING\_HASH: Encryption error
- 118** APP\_ERROR\_CREATING\_HASH\_DATA: Encryption error
- 119** APP\_ERROR\_DERIVING\_KEY: Encryption error
- 120** APP\_ERROR\_DECRYPTING\_DATA: Encryption error
- 121** APP\_ERROR\_BAD\_ENCRYPTION\_TYPE: Encryption error
- 122** APP\_ERROR\_ENCRYPTING\_DATA: Encryption error
- 123** APP\_ERROR\_CONVERTING\_DATA: Encryption error
- 124** APP\_ERROR\_AQUIRING\_ENHANCED\_PROVIDER: Encryption error
- 125** APP\_ERROR\_AQUIRING\_STANDARD\_PROVIDER: Encryption error
- 126** APP\_ERROR\_CREATING\_INIT\_VALUES: Initialization values could not be created
- 127** Unused
- 128** APP\_ERROR\_COULD\_NOT\_EXTRACT\_STRING: Encryption error
- 129** APP\_ERROR\_COULD\_NOT\_CREATE\_TRIAL\_STRING: Encryption error
- 130** Unused
- 131** Unused
- 132** Unused
- 133** Unused
- 134** APP\_INVALID\_PORT\_NUMBER: Invalid MySQL database port number. Must be greater than zero.
- 135** APP\_INVALID\_USER\_NAME: Invalid MySQL database user name.
- 136** APP\_INVALID\_DB\_PASSWORD: Invalid MySQL database password.
- 137** APP\_NO\_INTERNET\_CONNECTION: Could not connect to the internet
- 138** APP\_COULD\_NOT\_CREATE\_MACHINE\_ID: Could not create a machine identifier.

- 139** APP\_MISSING\_FIRST\_RUN\_DATE: Could not find the date the application was first run in evaluation mode
- 140** APP\_MISSING\_LAST\_RUN\_DATE: Could not find the date the application was last run in evaluation mode
- 141** APP\_INVALID\_FIRST\_RUN\_DATE: First run date found, but not valid
- 142** APP\_INVALID\_LAST\_RUN\_DATE: Last run date found, but not valid
- 143** APP\_NO\_APP\_NAME\_IN\_LICENSE: Could not find the application name in the license file
- 144** APP\_NO\_USER\_NAME\_IN\_LICENSE: Could not find a user name in the license file
- 145** APP\_NO\_ID\_VALUE\_IN\_LICENSE: Could not find an ID value in the license file
- 146** APP\_NO\_EXPIRATION\_DATA\_IN\_LICENSE: Could not find an expiration date in the license file
- 147** APP\_INVALID\_EXPIRATION\_DATA\_IN\_LICENSE: Could not find a valid expiration date in the license file
- 148** APP\_INVALID\_TRIAL\_STRING: Invalid trial string found
- 149** APP\_MISSING\_FIELD\_DATA\_1: Missing Field #1 in the evaluation string
- 150** APP\_MISSING\_FIELD\_DATA\_2: Missing Field #2 in the evaluation string
- 151** APP\_MISSING\_FIELD\_DATA\_3: Missing Field #3 in the evaluation string
- 152** APP\_MISSING\_FIELD\_DATA\_4: Missing Field #4 in the evaluation string
- 153** APP\_INVALID\_EVALUATION\_STRING: Invalid evaluation string
- 154** APP\_INVALID\_LICENSE\_DATA: Invalid/mismatched license file values
- 155** APP\_TRIAL\_READBACK\_FAILED: Trial data saved to the online database, but same data could not be read back (verification error)
- 156** APP\_FAILED\_TO\_GET\_TRIAL\_STRING: Failed to read the trial string from the online database
- 157** APP\_ERROR\_CONVERT\_FAILED: Failed to convert evaluation string
- 158** Unused

**159** Unused

**160** APP\_INVALID\_CLIENT\_FILE: Invalid SoftShell component detected. CRC's must match.

**399** APP\_ERROR\_UNKNOWN: The error cause could not be determined

## Binary Compatibility

When your application is compiled using a particular version of any ActiveX component, the class ID and interface ID of each object it uses will be compiled in. As long as those ID's do not change, the component remains "binary compatible". Upgraded versions of that component can be used without problem.

However, if a public member or method in the component is modified, a new class/interface ID will be created. Consequently, your application will not be able to find this new component as that reference no longer exists. For your application to work, the new component must be compiled in with your application.

## **Deactivate Registration ID**

The SoftShell™ Software Licensing System allows you to look for registration ID's in the online database Inactive table. If a registration ID is found in that table, your software can take the action you feel necessary.

The use of custom strings is a very useful tool for site licenses and other areas where some redistribution is encouraged. However, if someone were to post a copy of the registration number to a peer to peer service, usage of your product could become unlimited very quickly. To circumvent this problem, a "blacklist" feature has been added to the SoftShell client.

Please note – if you choose to use this feature, you must host your own online database.

## Installation & Distribution

Once you've developed your application for release, it will be time to decide how you want to distribute the files required for the SoftShell system. Although there are only three (3) main files that control the SoftShell system - SShllClient.dll, SShllInterface.dll, and SShllSrv.exe – these files have various runtime dependencies as well. On top of that, many of these files are COM components – files that must be registered to the system and the Service Control Manager.

Not to worry – there are several different options for you to choose from. These methods include:

1. Include the Bevan Engineering supplied redistributable files into your installation package. Allow the installation package to manage the MSI database, register components, and install the SoftShell service application SShllSrv.exe to the Service Control Manager (SCM).
2. Use the SoftShell Licensing Support installation package. Launching a single executable, all files will be installed and properly configured for you. In fact, an InstallShield © prerequisite file is available for you.
3. Using regsvr32.exe, use a batch file to copy and register files. Even the service application can be installed this way by passing a –i into the application. This command line parameter will install the service into the SCM.
4. Using regsvr32.exe, use a batch file to copy and register files. Instead of installing SShllSrv.exe as a service, pass a –h to the application. This will cause the application to run as a hidden application - not controlled by the SCM. It is still recommended you install the service into the system/system32 directory

## Files

The following files must be included in order for the SoftShell system to run:

SShllClient.dll

SShllInterface.dll

SShllSrv.exe

libmySQL.dll

mfc40.dll

msvcrt40.dll

VBMySQLDirect.dll

asycfilt.dll (included in Microsoft OLE 2.40 Merge Module)

oleaut32.dll (included in Microsoft OLE 2.40 Merge Module)

olepro32.dll (included in Microsoft OLE 2.40 Merge Module)

stdole2.tlb (included in Microsoft OLE 2.40 Merge Module)

comcat.dll (included in Microsoft Component Category Manager Library Merge Module)

msvcrt.dll (included in Microsoft C Runtime Library 6.0 Merge Module)

mdac\_typ.exe (included in Microsoft Data Access Components 2.8 Merge Module)  
msvbvm60.dll (included in Microsoft Visual Basic Virtual Machine 6.0 Merge Module)  
msvcp60.dll (included in Microsoft C++ Runtime Library 6.0 Merge Module)  
mfc42.dll (included in MFC 6.0 Merge Module)

For more details relating to installation and distribution, refer to the “SoftShell Installation Instructions” document available for download on the Bevan Engineering website [support](#) page.

## **NT Event Log**

Most errors and/or important events detected by the SoftShell service application (SShllSrv.exe) will be written to the Event log and can be viewed with the event viewer.

## Technical Support

Bevan Engineering provides free technical support via email. Address any technical issues to [support@bevanengineering.com](mailto:support@bevanengineering.com). Be sure to include the following information:

1. Product information and revision level.
2. Specific details of the problem. Screen shots could be helpful.
3. The operating system and service pack level.
4. The type of processor, amount of RAM, and other important information that could
5. be relevant.
6. If it's a development issue, the programming environment and service pack release.
7. The results after testing any sample applications

## Security Warning

No copy protection scheme is perfect. If someone wants to crack a copy protection system - no matter how complex - they will do it. Bevan Engineering has made every reasonable effort using the latest in encryption technology to prevent someone from getting around SoftShell's copy protection. However, Bevan Engineering has not and will not claim that the SoftShell copy protection system cannot be cracked and or circumvented.

## About

The SoftShell™ Software Licensing System was developed by Bevan Engineering, LLC. Bevan Engineering was founded in 2004 to provide highly specialized controls and software solutions for manufacturing facilities and OEMs. Many clients - including Fortune 500 companies - have been using Bevan Engineering products and services world-wide.

**All SoftShell™ Software Licensing System files developed by Bevan Engineering, LLC files are provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should this program or any other software developed by Bevan Engineering, LLC prove defective, you (not Bevan Engineering, LLC. or an authorized dealer) assume the entire cost of all necessary servicing, repair, or correction. In no event will Bevan Engineering, LLC. be liable to you for any damages , including any lost profits, lost savings, or other incidental or consequential damages arising out of the use or inability to use such program even if Bevan Engineering, LLC. or an authorized Bevan Engineering, LLC. personal computer dealer has been advised of the possibility of such damages, or for any claim by any other party.**

**No portion of the SoftShell™ Software Licensing System files may be reproduced. Unless specifically stated by your End Users License Agreement (EULA), licensed users may distribute the run-time version of these controls with their applications royalty free. Bevan Engineering, LLC. assumes no risk and/or liability with the redistribution of these files.**