How Does the AstroSubscription DLL works?

Part A – Desktop Side DLL(protectedDll)

1. the entry to this dynamic link library is through a constructor called “new”
2. the new constructor will be written inside a module where other functions are also stored, which would like to use. E.g. in the test protected DLL. We have a function call numbersquare and as the name suggest, it returns a square of a given number. Every time, when the function is accessed or referred from anywhere in the main program, the control is passed on to the constructor called new.
3. The “New” constructor has the following sequence of logic:
   1. it checks if the full license is valid. If the answer is, then the control is returned outside the constructor. Hence, all other functions are accessible for the user.
   2. In case the full license is not valid, then the next condition is checked. It checks whether subscription license is valid or not. If valid, then the subscription license is updated with change in the license key. The last part of the license key is the accessing data in time. The key is then, encrypted and the modified time is set to the current time.
   3. In case the subscription license is also not valid, then the program checks whether the trial license is valid or not. If the driver license is valid, then the license key is updated as in the case of subscription license.
   4. If the trial license is also invalid, then a message is displayed to renew or activate the license and it opens the activation form wherein manually. One is to enter the activation key to activate the license.
4. In the server side dynamic link library, the process is replaced by ultimate occasion and sending over Internet to the activation page internally to activate the license.
5. How does the activate function works?
   1. The entered activation key has the product ID in build inside.
   2. The product ID is extracted from the activation key.
   3. If the product ID is either for pearl or for Diamond then check for the subscription period.
   4. The validity is today plus the subscription period.
   5. Depending on the product ID, activate the license for trial or for subscription all for full.
6. The activate function refers to 3 different modules depending on the type of activation. In case of the full license the following process takes place. The activate function calls the class fulllicense.vb
7. the activate function has the parameter of activation key and it returns yes or no.
   1. Step number 1 is to generate a serial key through a function generates serial key with the parameters device ID, and “F”
   2. if the directory does not exist, it creates a directory called “AstroFolder”
   3. the activation key generated through a function generate activation key, which is parameters, the serial key and “F” as the product.
   4. If the generated key and the supplied activation key matches, then degenerated full key is then written inside the license path called licpath.
   5. The generateactivationkey, generateserialkey and deviceID all are part of the public class Astrolicense
   6. DeviceID is generated from processor ID and Hard disk ID
   7. adminkey\_ is a private variable and AdminKey is a public property
   8. Module AstroUtil available to all classes and it has the following functions
      1. getprocessorId()
      2. getHarddiskID()
      3. encode()
      4. reverse()
      5. public class rijndaelsimple
         1. public shared function encrypt\_
         2. public shared function decrypt\_
8. The subscription type activate refers to subscriptionlicense.vb
9. The activate function operated in the following way
   1. the product ID separated from the activation key.
   2. The generateactivationkey function has two parameters( serialisation key and the product ID)
   3. the trial key is a property which is decrypted form of the license key split into fields()
   4. it also needs the admin key internally.
   5. If the activation key supplied and the generated activation key matches. Then, it generates a license with trial key, subscription period and the product ID.
10. The Generate function has optional parameter key, optional subscription\_period as integer , productID and optional prefix as string
11. at the time of activation, the generate function is called as below:
12. Generate(trialkey, subscription., Product ID).
13. The trialkey is a property whose value at the beginning will be “” a null string.
14. In generate function it checks for the prefix and for the key. In the 1st call the key which is the decrypted license key being null, a new GUID key generated in 16 digits.
15. The expiry time is calculated as the current time added with the subscription period.
16. Plaintext = guid key, subsc\_period, productID, executivecreationtime, modified time, expirytime, currentTime
17. cipherText is rijandaelsimple.encrypt (plain text, other parameters)
18. the encrypted text is written inside the license file and the file creation time is set to the current time
19. on every access to the constructor “New” we check whether the license is valid or not for full, subscription, trial.
20. The Isvalid function for the subscription class is as follows:
    1. the 1st step is to check whether the license file exist if not clearly somebody has tampered it. The status will be returned as false
    2. the next step is to get the plaintext by decryptlicensetext() function.
    3. Get fields()
    4. Assign the fields(0) to Fields(6)
    5. Field(0) = guid key
    6. Field(1)= subscription\_period
    7. Field(2) = product\_id
    8. Field(3)= executive creation time
    9. Field(4) = Current time (Activation time) of the subscription
    10. Field(5)= expiry time (activation date + subscription period)
    11. Field(6)= current time (modified time) or access time
21. At this stage, it compares for discrepancies like license creation time must be = modified time (access time) or someone tampered then set status false and generate() from the scratch i.e. generate() without parameters.
22. Otherwise, show the remaining days and then update() the license with the field(6) replaced with the modified time or the access time. Here the call to generate(Prefix:=Prefix) is with the prefix if we are still not reached the expiry date
23. The generated key with the last parameter changed is then encrypted and written in the license file and the file access time is set as the modified time, so that when we compare the next time the license creation time externally and inside the key it must match.
24. The update() function is complete for the subscription
25. The same logic is for the Trial period except the trial period is fixed and passed the value e.g. 7 days.
26. The full key logic is different and uses the fulllicense class which is more simple to understand.