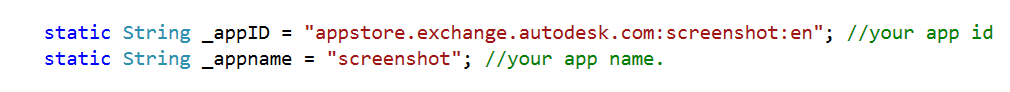
**Entitlement API: sample project for AutoCAD (AutoCAD verticals) desktop Apps.**

The sample code has two projects.

1. .NET AutoCAD plug-in, which represent an App in the exchange store. (Except for command “PrintlicenseInfo”, this is written for testing purpose) – **used by users of the App**
2. .NET application “LicGenUtility”, for generating license – **used by publishers.**

For testing the use of .NET AutoCAD plug-in & LicGenUtility:

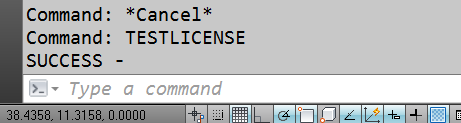
Download the sample App screenshot from exchange store <https://apps.exchange.autodesk.com/ACD/en/Detail/Index?id=appstore.exchange.autodesk.com%3ascreenshot%3aen> . This sample code uses screenshot application id to check for entitle service.



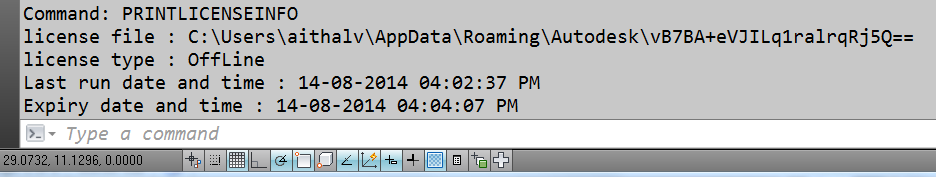
Build and load the ADNtestApp sample application.

Run command “Testlicense”.

Command checks the license and prints the status. The logic followed to checking as explained in flow chart.



Run command “PrintlicenseInfo”. This commands prints the license details. This command is written for reading the local license file.

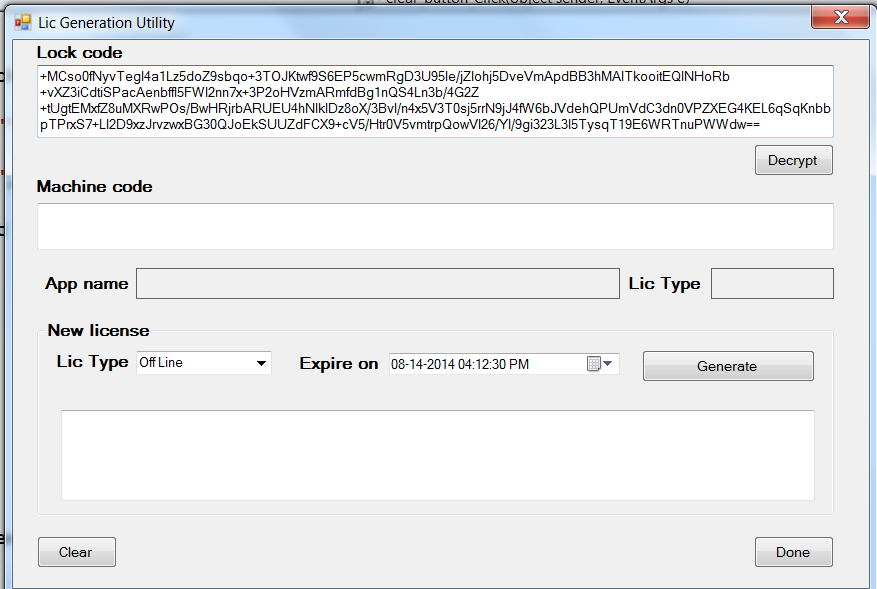


To get offline license or to convert the license from offline to online or vice versa, user of the exchange store App needs to send the machine details to the publishers. For this “getLockcode” command is provided in the ADNtestApp sample. This command reads the network ports and creates an encrypted string. This string is shown in AutoCAD inside a dialog. User of the App needs to send this code to publisher/App developer.



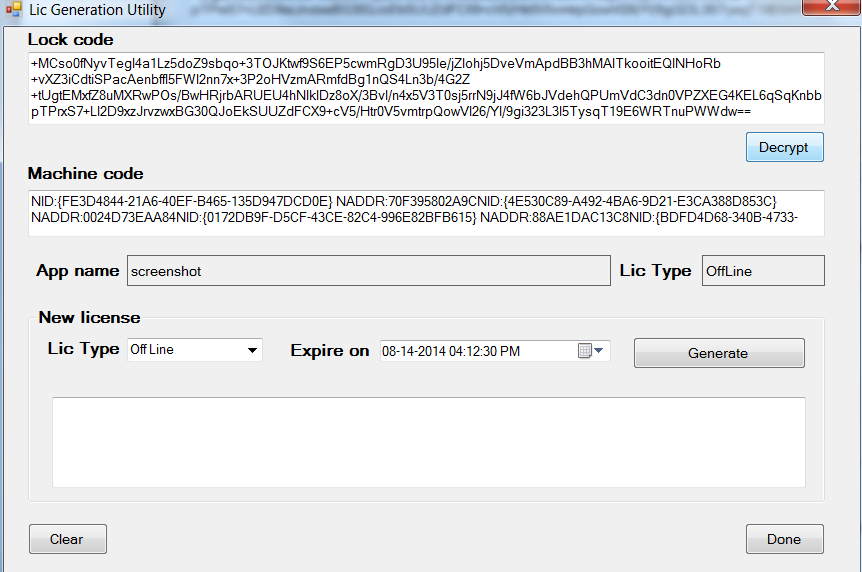
Build and run the LicGenUtility application.

Copy the code provided by “getLockcode” command in “lockcode” section of the LicGenUtility tool as shown

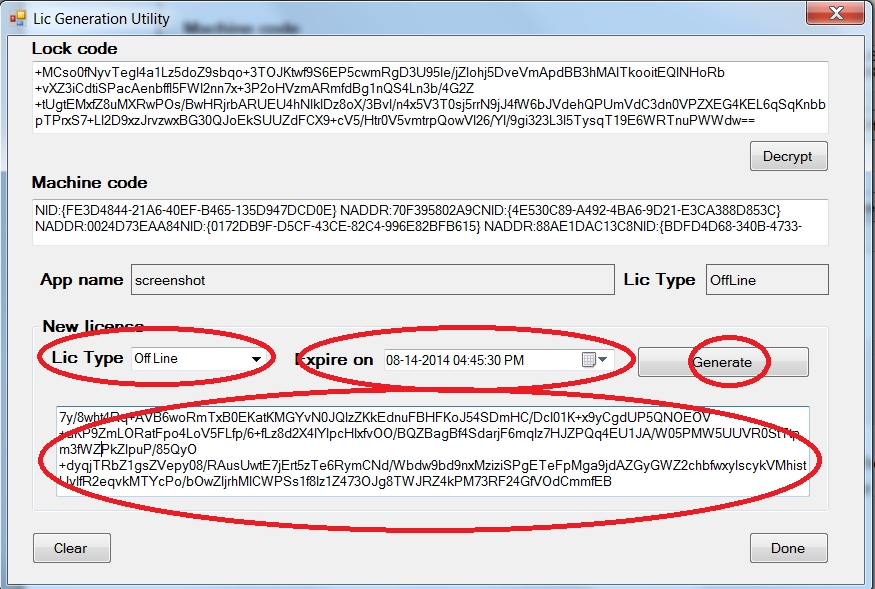


Press Decrypt button

The code is decrypted and the machine code, app name and license is populated in the dialog box as shown below



Now, use license type combo box and expire date options set the required values for the new license and press the button “generate” . This will create a new license text, which needs to be send to App user



In AutoCAD, user needs to issue command “updatelicense” implemented in ADNtestApp app. This shows a dialog box, where users’ needs to copy the license text generated by LicGenUtility.

This step will update the local license file with options you have selected in the LicGenUtility.

**Entitlement API: for AutoCAD (AutoCAD verticals) desktop Apps.**

When run, an App should check with exchange store server if the user is a legitimate user or not, if the user is online. If user is not online, then App should check a file in the system to identify how long since the app last contacted the exchange store server. If user has not contacted the exchange store server for certain number of days (like 15 or 30), App should warn the user and stop working until the user goes online and re-authenticates.

When a user downloads the App from the store, the user id is stored in the exchange store server. Publishers can query and verify these ids at runtime. Refer to this blog post - <http://adndevblog.typepad.com/cloud_and_mobile/2014/05/how-to-protect-my-intellectual-property-of-my-app-on-autodesk-exchange-part-3.html> . Publishers can use this id to identify whether the user is a legitimate user (a user who has downloaded App) or not. However, there are some circumstances where a new user may not be able to immediately ‘authenticate’ the App, such as:

* The user is running the app for the first time when they don’t have an internet connection available.
* The App was purchased as part of a ‘bulk purchase’ – in which case the user running the app is not going to be the same user who bought the multiple copies of the app.

To allow for such situations, the App should continue to work for certain time (15 or 30 days) as a fully functional trial, even if the user is offline. To support this offline validation, publishers need save few values in local hard-disc of the user.

Read license file, create the file if required

License mode type

Start

Does Machine ID matches?

Is ONLINE USERID Present?

Is User entitled for the app?

Does Machine ID matches?

System date back dated?

Time since last online < 30 days?

IS Date expired?

Updated license file

Success: Allow the App to work..

Online

Offline

no

YES

Yes

Yes

No

No

Yes

No

No

Yes

Yes

Yes

Yes

No

System date back dated?

No

No

END

Show error message and exit

How to achieve the objective:

1. Read the content of license file (see below about location & content of the license file)
2. If license file not found, then create the license file in runtime with default values. By default license type will be online.
3. If license found, then check if the license mode is offline or online.

**Online mode:**

1. Read the value of the “ONLINEUSERID” system variables.
2. If ONLINEUSERID found, use the Exchange Entitlement API (as explained the blog) to verify the user.
3. If user is legitimate, update the information in offline file and allow the command to continue.
4. If user is not legitimate, show an appropriate message and either disable or restrict the Apps functionality.
5. If “ONLINEUSERID” returns null, then check for machine id.
6. Check for back dating.
7. Check for last date when user had contacted the exchanges store. If the last contact is less than the certain number of days, allow the user to use the App.
8. If step 5 or 6 or 7 fails, then show the error message and restrict the use of the App.

**Offline mode:**

1. Verify the machine id.
2. Verify for back dating.
3. Verify for expiry date of the App – this could be perpetual.
4. If step 1 or 2 or 3 fails, then show the error message and restrict the use of the App

**What are stored in license file?**

1. AppID – Id which is same as exchange store App id.
2. LicensingMode – Online or Offline
3. MachineLockCode – network card id.
4. StartDateTimeStamp
5. LastRunDateTimeStamp
6. ExpiryDateTimeStamp

In Online mode, update - AppID, MachineLockCode = this machine, StartDateTimeStamp = current date, LastRunDateTimeStamp = current date, ExpiryDateTimeStamp = current date + 30 days.

In offline mode update: Only update LastRunDateTimeStamp = current date.

**Offline license file name : <AppId encrypted>.lic**

**Offline license file location:** Any per-user location where user has read/write permission. This decision is taken by publisher. Like appdata folder.

**License for machines which are always offline:**

1. By default the App will work for 15 or 30 days.
2. On running of command (example : RegisterMyApp), a prompt is shown with machine id (encrypted)
3. User is needed to contact publisher with the encrypted machine id & the App name for which user needs the offline license.
4. Publishers then uses machine id, App id and creates an encrypted license string.
5. This encrypted license string is passed to the user.
6. User needs to input this license string in the App. Then App decrypts this license string and makes required entries in the licensing file.
7. Publishers are needed to develop a simple License Generator utility to perform above task.

**A D N will provide simple samples demonstrating above tasks. Publishers need to customize the samples to suite their requirement.**