



# شناسه و قفل سخت افزاری نوین

## Novin USB Token and USB Dongle



- کارت شناسایی و امضای دیجیتال
- پرونده الکترونیک
- مدیریت کلمات عبور
- ورود به سیستم عامل
- محدودیت دسترسی به امکانات سیستم عامل
- رمز نگاری و رمز گشایی اطلاعات و فایل ها
- قفل سخت افزاری جهت حفاظت از نرم افزار در مقابل کپی غیر مجاز

((نگارش ۵))

زمستان ۱۳۸۵



| صفحه | فهرست مطالب   |
|------|---|
| ۴    | مقدمه.....  |
| ۴    | ۱- مشخصات کلی شناسه نوین.....                       |
| ۴    | ۱-۱- شماره سریال شناسه.....                         |
| ۴    | ۱-۲- کد اختصاصی مشتری (VID: Vendor ID).....         |
| ۵    | ۱-۳- رمز عبور مدیریت (Admin Password).....          |
| ۵    | ۱-۴- رمز عبور برنامه نویس (Developer Password)..... |
| ۵    | ۱-۵- رمز عبور کاربر (User Password).....            |
| ۶    | ۱-۶- شماره نرم افزار یا PID (Product ID).....       |
| ۶    | ۱-۷- کلیدهای رمزنگاری AES.....                      |
| ۶    | ۱-۸- مشخصه تعداد کاربران شبکه (NET).....            |
| ۶    | ۱-۹- تنظیمات مربوط به وب.....                       |
| ۶    | ۱-۱۰- محدود کننده استفاده از شناسه.....             |
| ۷    | ۱-۱۱- بخش داده آزاد.....                            |
| ۷    | ۱-۱۲- تنظیمات سطوح دسترسی برای بخش داده.....        |
| ۷    | ۲- نصب راه انداز شناسه.....                         |
| ۸    | ۳- نحوه استفاده از شناسه.....                       |
| ۸    | ۳-۱- آشنایی با نرم افزار برنامه ریز.....            |
| ۹    | ۳-۱-۱- نوار ابزار بالا.....                         |
| ۱۰   | ۳-۱-۲- نوار وضعیت پایین.....                        |
| ۱۰   | ۳-۱-۳- تب تنظیمات کلی (General Options).....        |
| ۱۳   | ۳-۱-۴- تب شمارنده/زمانسنج (Timer/Counter).....      |
| ۱۴   | ۳-۱-۵- تب قسمت داده (Data Section).....             |



- ۱۶-۱-۳- تب تنظیمات سطوح دسترسی قسمت داده (Data Permissions).....
- ۱۷-۱-۳- تب تنظیمات شبکه و وب (Net and Web options).....
- ۱۷-۱-۳- برنامه ریزی شناسه.....
- ۱۹-۱-۳- ایجاد شناسه های مشابه.....
- ۲۰-۲-۳- ارتباط با شناسه در برنامه نویسی.....
- ۲۱-۲-۳- متدهای رابط سری A.....
- ۲۸-۲-۳- متدهای رابط سری B.....
- ۳۷-۲-۳- متدهای رابط سری C.....
- ۳۸-۲-۳- متدهای رابط در شبکه.....
- ۳۹-۲-۳- انواع خطاهای بازگشتی.....
- ۴۰-۴- کاربردهای شناسه نوین.....
- ۴۰-۱-۴- قفل سخت افزار.....
- ۴۱-۲-۴- قفل سخت افزاری تحت شبکه.....
- ۴۴-۳-۴- شناسه دیجیتال تحت وب.....
- ۴۴-۱-۳-۴- ارتباط با صفحه کنسول.....
- ۴۶-۲-۳-۴- روش OTP.....
- ۴۷-۴-۴- امضای دیجیتال.....
- ۴۷-۵-۴- مجوز ورود به ویندوز.....



## مقدمه

کپی کردن غیر مجاز نرم افزارها یکی از بزرگترین مشکلات توسعه دهندگان نرم افزار در جهان بخصوص در ایران می باشد و یکی از بهترین، مطمئن ترین و با صرفه ترین راهها برای جلوگیری از آن، استفاده از قفل های سخت افزاری می باشد.

شرکت نوین افزار که خود یکی از تولید کنندگان نرم افزار می باشد، با توجه به کارآمد نبودن اکثر قفل های موجود در بازار ایران و همچنین گران بودن نمونه های خارجی، بر آن شد تا با توجه به این دو موضوع قفلی کارا با هزینه ای ارزان و به صورت داخلی تولید و به توسعه دهندگان نرم افزار عرضه نماید.

پس از رسیدن به تولی انبوه قفل سخت افزاری و مطالعه روی بازار مصرف این کالا بر آن شدیم که کاربردهای بهتری علاوه بر قفل سخت افزاری برا محافظت از نرم افزار پیدا کنیم. طی این مرحله موفق شدیم قفلهایی ارائه کنیم که در شبکه و وب قابل شناسایی باشند. در بحث شبکه اشتراک گذاشتن یک قفل و محدود کردن تعداد کاربران مورد هدف قرار دارد، همچنین در بحث وب شناسایی کاربران یک وب سایت در اینترنت هدف اصلی کار قرار دارد.

## ۱- مشخصات کلی شناسه نوین

شناسه های ساخت این شرکت علاوه بر پارامترهای درونی فراوان، دارای الگوریتم رمزنگاری AES نیز می باشد که یکی از جدیدترین الگوریتم های رمزنگاری دنیا می باشد، که تاکنون شکسته نشده است. الگوریتم استفاده شده در شناسه با استفاده از کلید ۱۲۸ بیتی (تعداد حالات ممکن برای کلیدها  $3.40 \times 10^{38}$ ) عمل می کند که می توان گفت که امنیت استفاده از شناسه تقریبا برابر امنیت همین الگوریتم رمزنگاری است. برای بهتر شناختن شناسه به پارامترهای درونی شناسه اشاره می کنیم:

### ۱-۱- شماره سریال شناسه

این عدد توسط شرکت سازنده به شناسه اختصاص می یابد و ساختاری شبیه آدرس IP در شبکه را دارد. شماره سریال یک عدد یکتا است که می توان از آن در شناسایی کاربران انتهایی (End-User) از آن استفاده کرد.

### ۱-۲- کد اختصاصی مشتری (VID: Vendor ID)

برای هر استفاده کننده یعنی افرادی که شناسه خام تهیه می کنند، یک کد اختصاصی یا VID توسط شرکت سازنده بطور رایگان تخصیص داده می شود و در تمام مراحل استفاده از شناسه این کد لازم است، این کد از چهار عدد بین (۰-۲۵۵) که با نقطه از هم جدا شده اند (شبیه به آدرس IP در شبکه) تشکیل شده است. این کد جزو سطح امنیت شناسه ها به حساب آمده و فقط توسط شرکت سازنده قابل اختصاص است. پس از دریافت اولین سری از شناسه های خریداری شده، در مراحل بعدی برای دریافت شناسه های خام با همان VID قبلی مدارک





## ۱-۶- شماره نرم افزار یا PID (Product ID)

شماره نرم افزار نیز مانند VID از چهار عدد تشکیل شده است که با نقطه از هم جدا شده اند. از این کد می توان برای شماره گذاری نسخه های نرم افزارها استفاده کرد. برای تنظیم آن باید VID و رمز عبور admin لازم است ولی برای گرفتن آن بهتر است از رمز developer استفاده کرد.

## ۱-۷- کلیدهای رمزنگاری AES

همانطور که قبلا ذکر شد یکی از مهمترین قابلیت های موجود در شناسه های نوین الگوریتم های رمزنگاری AES است که در حال حاضر از جمله الگوریتم های برتر دنیا می باشد. الگوریتم رمزنگاری AES براساس کلید ۱۲۸ بیتی عمل می کند و کلیه مقادیر ورودی را با استفاده از این کلید، رمزنگاری و رمزگشایی می کند. در شناسه نوین برای انجام عمل رمزنگاری ۲ کلید وجود دارد، کلید اول (کلید همگانی) توسط admin تنظیم شده و معمولا بین تمام شناسه های ارائه شده از سوی وی ثابت می باشد. کلید دوم (کلید شخصی) توسط User فقط یکبار تنظیم می شود و دیگر قابل تغییر نیست. برای بالابردن ضریب اطمینان این کلیدها در داخل شناسه ذخیره می گردد و عملیات های رمزنگاری و رمزگشایی درون شناسه و با استفاده از این کلید انجام می پذیرد. و هیچ سرویسی برای بازیابی کلید داخلی وجود ندارد. این کلیدها از ۱۶ عدد (بین ۲۵۵-۰) که با نقطه از هم جدا شده اند تشکیل شده اند.

## ۱-۸- مشخصه تعداد کاربران شبکه (NET)

در صورتی که بخواهیم شناسه در شبکه به اشتراک گذارده شود، باید حداکثر تعداد کاربران مجاز استفاده کننده از شناسه را با استفاده از این پارامتر تعیین کرد. این عدد می تواند یکی از اعداد ۱ تا ۲۵۵ بوده یا در حالت بینهایت (Unlimited) تنظیم شده باشد.

## ۱-۹- تنظیمات مربوط به وب

این بخش از ۴ یا ۵ کلمه تشکیل می شود که در مجموع می توانند ۷۶ کاراکتر باشند. این کلمات پارامترهای مربوط به صفحه کنسول در وب سایت را در خود ذخیره می کند و توضیحات آن در بخش ۳۳۳۳۳۳ توضیح داده شده است.

## ۱-۱۰- محدود کننده استفاده از شناسه

در این قسمت می توان عددی ۳ بیتی (حداکثر ۱۶,۷۷۷,۲۱۵) را به عنوان حداکثر مقدار شمارنده تعیین کرد، این عدد را می توان در مراحل مختلف استفاده توسط متد مربوطه کاهش داد یا با استفاده از فعال کردن زمانسنج از به طور خودکار از هر ۵,۵۹ ثانیه یک واحد کاهش داد. در صورتی که در مراحل فوق مقدار به صفر برسد شناسه به حالت suspend می رود. برای خارج کردن شناسه از حالت suspend می توان شناسه یا فقط قسمت Timer/Counter را با استفاده از سطح دسترسی admin ریست کرد. روش دیگری که برای اینکار مخصوصا از



راه دور مناسب است، استفاده از Activation Code است. این رشته که یک رشته ۳۲ کاراکتری است به دو روش قابل تولید است. هنگامی که شناسه در اختیار admin باشد با استفاده از سطح دسترسی admin قابل ایجاد است. علاوه بر این با داشتن شماره سریال شناسه، با استفاده از شناسه دیگری که کلید AES اول آنها با هم یکسان است نیز قابل تولید است.

### ۱-۱۱- بخش داده آزاد

این بخش با توجه به فعال یا غیرفعال بودن تنظیمات وب می‌زان ۳۲۰ یا ۴۰۰ بایت حافظه قابل خواندن و نوشتن را در اختیار قرار می‌دهد. در صورتی که به صورت Unicode از این بخش استفاده کرد مقدار حافظه به صورت ۱۶۰ یا ۲۰۰ کاراکتر در می‌آید.

### ۱-۱۲- تنظیمات سطوح دسترسی برای بخش داده

بخش داده ۴۰۰ بایتی شناسه را می‌توان به حداکثر ۶ بخش تقسیم کرد و برای هر بخش سطح دسترسی لازم برای نوشتن یا خواندن را تعیین کرد. تنظیمات برای ۵ بخش اول قابل اعمال است و بخش ششم مازاد بخش داده ای باشد که ممکن است در ۵ بخش اول استفاده نشده باشد. برای بخش آخر به صورت پیشفرض سطح دسترسی (۱،۱) تعلق می‌گیرد.

سطوح دسترسی برای خواندن و نوشتن نیز در ۴ سطح مورد بررسی قرار می‌گیرد.

سطح صفر: همگانی (everyone) که در دسترس همه قرار دارد.

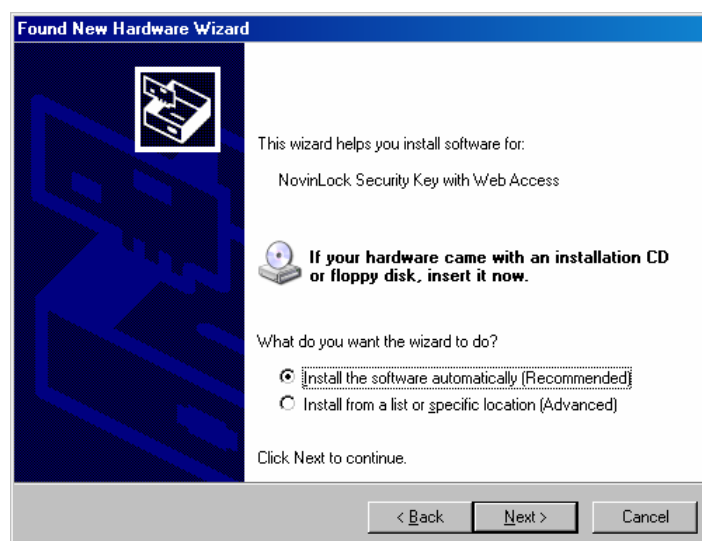
سطح یک: برای برنامه نویس که توسط سطح دسترسی admin و developer قابل دسترس است

سطح دو: برای مدیر که فقط توسط سطح دسترسی admin قابل دسترسی است

سطح سه: برای کاربر که فقط در دسترس User می‌باشد

## ۲- نصب راه انداز شناسه

در اولین اتصال شناسه به کامپیوتر باید درایور شناسه نصب شود.





برای این منظور بعد از اتصال شناسه به پورت USB پنجره Found New Hardware نمایان می‌شود. در این حالت CD مربوط به شناسه را در درایو قرارداده و گزینه Next را انتخاب نمائید، در حین نصب، پنجره ای نمایان می‌گردد، گزینه Continue Anyway را انتخاب نمائید، درایو شناسه بطور کامل نصب می‌گردد.

از این به بعد در هر اتصال، آیکون مربوطه در سمت راست پائین صفحه در

روی نوار ابزار ویندوز نمایان می‌گردد.



### ۳- نحوه استفاده از شناسه

برای ارتباط و استفاده از شناسه در نرم افزارهای مختلف باید ابتدا اطلاعات لازم در شناسه نوشته و سپس روتینهای مربوط به شناسایی و ارتباط با شناسه فراخوانی شوند. برای شناسایی و نوشتن اطلاعات اولیه در شناسه باید از نرم افزار برنامه ریزی شناسه (NLBuilder) استفاده می‌شود. همچنین ارتباط نرم افزار با شناسه از طریق یک فایل کتابخانه ویندوز (DLL) برقرار می‌گردد.

#### ۳-۱- آشنایی با نرم افزار برنامه ریز

این نرم افزار جهت آماده سازی شناسه، قبل از استفاده در نرم افزار می‌باشد. به این صورت که بعد از تهیه شناسه باید اطلاعات لازم توسط این نرم افزار در شناسه ثبت گردد و سپس این اطلاعات با استفاده از روتین های موجود، از داخل سورس برنامه مورد ارزیابی قرار گیرد. این نرم افزار در شاخه Programmer در CD همراه شناسه موجود است.



کلیه مراحل برنامه ریزی اصلی جهت استفاده به عنوان شناسه سخت افزاری در تب General Options انجام می پذیرد.

### ۳-۱-۱- نوار ابزار بالا

در قسمت بالای صفحه یک نوار ابزار موجود می باشد که شامل آیکون های زیر می باشد.

جدید: برای ایجاد یک پروژه جدید استفاده می شود، انتخاب این گزینه باعث حذف شدن تمام مقادیر وارد شده می گردد و صفحه آماده دریافت اطلاعات جدید می گردد.

فراخوانی: در صورتیکه پروژه ای را قبلاً ذخیره کرده باشید با استفاده از این گزینه می توانید دوباره پروژه را فراخوانی کنید، ۲ نوع فایل قابل فراخوانی می باشد.

- XML: که یک فرمت استاندارد می باشد، مزیت استفاده از این فرمت اینست که می توانید اطلاعات را در نرم افزارهای دیگر استفاده کنید و از معایب آن اینست که امنیت آن پائین است و به راحتی قابل خواندن است.

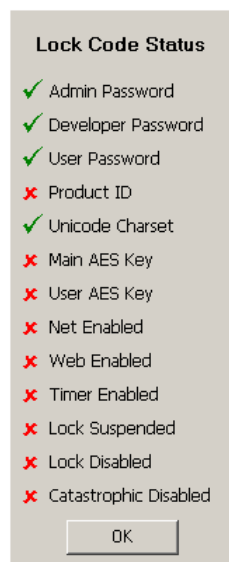
- NLD: این فرمت اختصاصی می باشد و بصورت رمز نگاری شده و با رمز عبور می باشد، به این صورت که در موقع ذخیره دو رمز عبور دریافت می شود یکی Administrator و دیگری User. در موقع فراخوانی مجدد این فایل کلمات عبور سوال می شود.

ذخیره: اطلاعات وارد شده را در فایل ذخیره می نماید تا در مواقع لازم بتوان پروژه را بازیابی نمود.

دو نوع فایل قابل بازیابی می باشد که در قسمت قبل در مورد آن توضیح داده شد فقط در مورد فایل های NLD در موقع بازیابی رمز عبور درخواست می شود، در صورتیکه با رمز عبور مربوط به administrator وارد شوید به کلیه اطلاعات دسترسی خواهید داشت و می توانید اطلاعات را مشاهده و یا تغییر دهید، اما در صورتیکه با رمز عبور مربوط به Users وارد شوید اطلاعات قابل نمایش نخواهد بود و فقط می توانید اطلاعات را در شناسه ذخیره نمائید، این قابلیت امکان ریزی شناسه توسط کارمندان عادی را فراهم می نماید.

بازخوانی شناسه: با زدن این کلید اطلاعات شناسه (از قبیل شماره سریال و ...) وصل شده به کامپیوتر دوباره در نرم افزار بارگذاری می شود.

نمایش وضعیت شناسه: این کلید به شما امکان می دهد که بتوانید از تنظیم شدن تک تک امکانات شناسه آگاه شوید یا اگر شناسه از کار افتاده باشد وضعیت آن را گزارش می دهد. پس از زدن این کلید پنجره ای مانند شکل مقابل نمایش داده می شود.



نوشتن اطلاعات در شناسه: در صورت تکمیل بودن اطلاعات با استفاده از این آیکون می توان اطلاعات را در شناسه ذخیره نمود. این عمل را می توان به صورت متناوب تکرار کرد به این صورت که بعد از رایت شدن شناسه، شناسه را تعویض و با انتخاب این




گزینه عمل رایت را دوباره انجام داد .

 Reset: شناسه را Reset می‌کند و اطلاعات آن را به حالت اولیه بر می‌گرداند.

دقت داشته باشید که بعد از این عمل اطلاعات موجود در صفحه نمایش تغییر نخواهد کرد، برای دیدن اطلاعات موجود در شناسه باید حتما اطلاعات از شناسه خوانده شود.

توجه داشته باشید که برای انجام عمل Reset باید حتما رمز عبور admin و VID در قسمت های مربوط وارد شوند و بدون ورود، عمل Reset مقدور نمی باشد.

 خروج: برای خروج از نرم افزار باید این آیکون را انتخاب کرد.

### ۳-۱-۲- نوار وضعیت پایین

در این قسمت می‌توانید اطلاعات شناسه شناسایی شده از قبیل شماره سریال ، سائز کارکتری حافظه موجود و فرمت داده را مشاهده کنید. هنگامی که نگارش شناسه با نرم افزار برنامه ریز سازگار نباشد پیغام Invalid Version در این قسمت مشاهده می‌گردد.

### ۳-۱-۳- تب تنظیمات کلی (General Options)

این تب از بخشهای مختلف زیر تشکیل شده است:

۱- قسمت وارد کردن VID (کد ویژه مشتری)

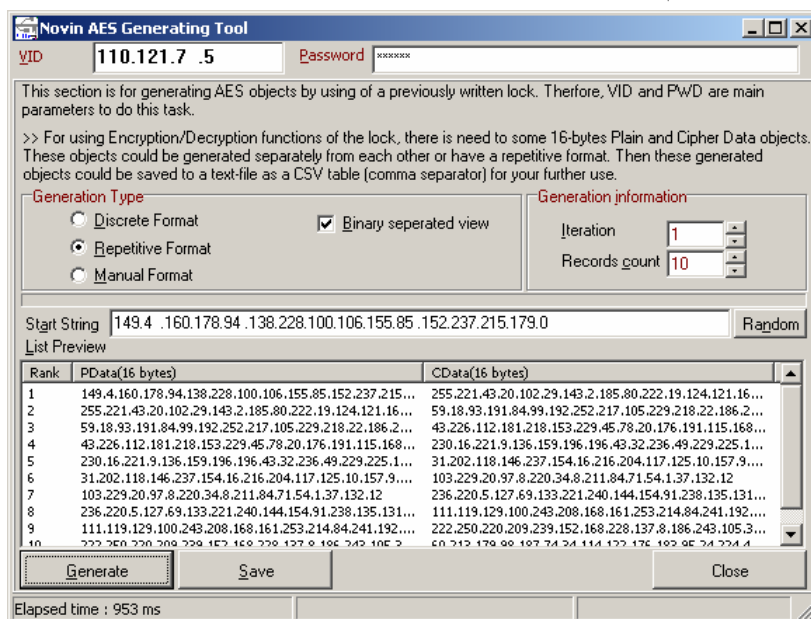
۲- تنظیمات رمز های عبور. در باکس اول این بخش باید رمز عبور سطح اول (Admin) فعلی شناسه را برای انجام عملیات برنامه ریزی وارد کنید. همچنین از طریق سایر گزینه های می‌توانید رمز عبور سطح اول و دوم را تغییر داد، یا رمز عبور سطح سوم (User) را به خالی ریست کرد.



۳- تنظیم کد محصول (Product ID) از این طریق امکانپذیر است.

۴- تنظیم نوع رشته در قسمت داده برای حالت Ansi یا Unicode از این قسمت انجام می‌شود. توجه داشته باشید که زمانی که در قسمت داده شناسه زبان غیر لاتین استاندارد مانند فارسی استفاده می‌کنید حتماً از حالت Unicode استفاده کنید.

۵- این قسمت برای تنظیم کلید اصلی (همگانی) AES استفاده می‌شود یعنی همان کلید اول. از طریق کلید AES Generator در این قسمت می‌توانید به تولید جفت کد های Plain و Cipher و ذخیره آن در فایل متنی اقدام کنید. با ورود به این بخش در بالای صفحه محلی برای ورود VID و رمز عبور مشاهده خواهید کرد که در صورت وارد کردن آنها در خود نرم افزار برنامه ریز به صورت خودکار در این قسمت نمایش داده می‌شوند.



موارد زیر در نرم افزار دیده می‌شوند:

Generation Type: در این قسمت چهار گزینه موجود است در قسمت سمت چپ باید یکی از سه گزینه

انتخاب شود این گزینه ها عبارتند از:

- Discrete Format: در صورت انتخاب این گزینه اعداد بصورت تصادفی تولید و عمل رمز گذاری انجام می‌شود.
- Repetitive Format: در صورت انتخاب این گزینه ، اولین مقدار از قسمت Start String انتخاب می‌گردد و سپس عمل رمز گذاری بر روی آن انجام و جواب بدست آمده دوباره به عنوان ورودی بعدی استفاده می‌گردد و مقادیر بدست آمده به لیست اضافه می‌گردد.
- Manual Format: در صورتیکه بخواهید اعداد مورد نظر خود را وارد و سیستم عمل رمز نگاری را برای شما انجام دهد، از این گزینه استفاده می‌کنید در این حالت در هر بار تکرار عمل باید مقدار مورد نظر را در Start String وارد کرد ، بعد از رمز نگاری مقدار وارد شده و رمز شده آن به لیست اضافه می‌شود.



- Binary Separated View: در صورتیکه این گزینه انتخاب شده باشد کلیه مقادیر تولید شده که همگی ۱۶ بیتی می‌باشد، بصورت ۱۶ عدد بین ۰ تا ۲۵۵ که با نقطه از هم جدا شده اند (DNString) نمایش داده می‌شوند و در غیر اینصورت بصورت رشته نمایش داده می‌شوند.

Generation Information: در این قسمت باید دو مقدار برای تولید و رمز گشائی تعیین شود.

- Iteration: همانطور که قبلا گفته شد در موقع انجام عمل رمز نگاری می‌توانید این عمل را چند بار تکرار کنید بطور مثال در صورتیکه این عدد را سه انتخاب نمائید در هر بار عمل رمز نگاری سه بار تکرار می‌شود به این صورت که بعد از رمز نگاری، مقدار حاصل شده دوباره رمز نگاری و مقدار بدست آمده برای بار سوم رمز نگاری و در نهایت مقدار حاصل شده نمایش داده می‌شود، و برای رسیدن به عدد اول باید عمل رمز گشائی نیز سه بار انجام شود.

- Record Count: این عدد تعداد اعداد تولید شده را مشخص می‌کند بطور مثال در صورتیکه عدد ۱۰۰ را وارد کنید، در پایان ۱۰۰ عدد تولید و رمز شده آن محاسبه و در خروجی چاپ می‌شود.

Start String: موارد کاربرد در قسمتهای قبل توضیح داده شد در صورت تمایل می‌توان با استفاده از کلید

Random مقادیر تصادفی تولید و در این قسمت قرار داد.

List Preview: کلیه اعداد تولید شده در این لیست که در قسمت پائین صفحه می‌باشد، نمایش داده

می‌شوند.

با زدن کلید سمت راست موس، منویی شامل دو گزینه نمایش داده می‌شود:

Clear All: باعث حذف تمام مقادیر موجود در لیست می‌گردد.

Delete: رکورد جاری در لیست را حذف می‌کند.

Generate: انتخاب این کلید عمل رمز نگاری را آغاز می‌کند و مقادیر بدست آمده در لیست اضافه می‌شوند.

Save: انتخاب این کلید مقادیر تولید شده و موجود در لیست را در یک فایل متنی ذخیره می‌کند، این فایل

بصورت Comma Separator می‌باشد.

جواب های بدست آمده را می‌توانید در لابلای اطلاعات و یا بصورت یک فایل باینری ذخیره نمائید و در

قسمت های مختلف برنامه بصورت تصادفی یکی از عددها را انتخاب و بعد از ارسال به شناسه و بدست آوردن

جواب، آن را با داده های موجود مقایسه نمائید و از بودن شناسه مطمئن شوید.

Close: باعث خروج از صفحه جاری می‌شود.



### ۳-۱-۴- تب شمارنده/زمانسنج (Timer/Counter)

The screenshot shows the 'Novin Lock Builder (NOVIN AFZAR CO.)' application window. The 'Timer/Counter' tab is active. It contains the following elements:

- A section titled 'Enter Maximum Time that the lock will work (MAX=1805 Days)' with a checkbox for 'Enable Timer (Automatically Decrease Value every 5,59 Sec)'. Below this are three spinners for 'Days', 'Hours', and 'Minutes', all set to 0.
- A section titled 'Enter Maximum Counter Value (MAX=16777216)' with a text input field containing '0' and 'Get', 'Set', and 'Reset' buttons.
- A section titled 'Generate Activation Code for Reset Timer/Counter remotely' with radio buttons for 'Current Device Serial' (selected) and 'Manual Serial Number'. A text input field contains '5.110.110.110' and a 'Generate Activation Code' button.
- A status bar at the bottom reads: 'Device Ready( SN : 5.110.110.110 , MemorySize : 200 (Unicode))'.

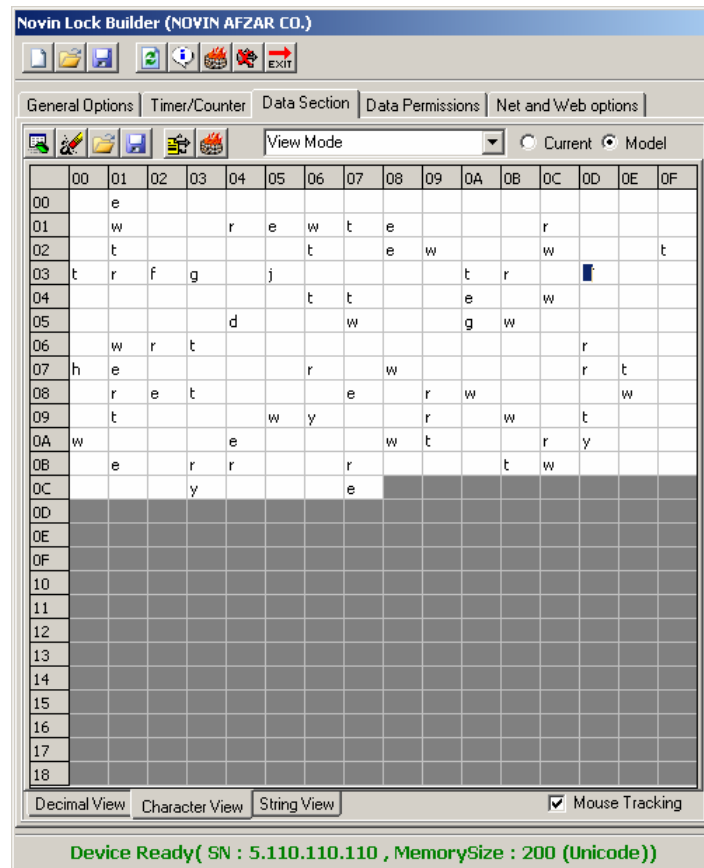
این قسمت زمانی استفاده می‌شود که می‌خواهید شناسه شما برای دفعات یا زمان محدود قابل استفاده باشد که برای نگارشهای آزمایش (Demo) مناسب می‌باشد. داخل شناسه زمانسنجی وجود دارد که از هر ۵,۵۹ ثانیه یک واحد از مقدار شمارنده را کم می‌کند. در صورتی که مقادیر مربوط به روز، ساعت یا دقیقه را تغییر دهید مقدار معادل شمارنده در باکس مربوطه محاسبه و نوشته می‌شود. چنانچه گفته شد می‌توان کاهش این عدد را به عهده زمانسنج قرار داد یا به صورت غیرخودکار از طریق متد مربوطه آن را کاهش داد. زمانی که این عدد به صفر برسد شناسه به حالت Suspend می‌رود. و خروج از این حالت به دو روش انجام می‌پذیرد.

۱- استفاده از کلید Reset در همین تب بعد از وارد کردن VID و Admin Password در تب مربوطه.

۲- ایجاد Activation Code با استفاده از داشتن شماره سریال شناسه مورد نظر و استفاده از خود شناسه یا شناسه‌ی که کلید AES اول آنها با هم برابر باشند و استفاده از Activation Code ایجاد شده در متد فعالسازی مربوطه.



## ۳-۱-۵- تب قسمت داده (Data Section)



این قسمت که می‌تواند برای خواندن و نوشتن قسمت داده شناسه استفاده شود، بخش داده شناسه را در سه وضعیت نمایش می‌دهد:

- **Decimal View:** در این حالت عدد ذخیره شده در هر خانه بصورت دسیمال نمایش داده می‌شود، بطور مثال در ابتدا، حافظه شناسه خالی می‌باشد و تمام خانه‌ها با عدد ۳۲ پر شده‌اند.
- **Character View:** در این حالت مقدار هر خانه به صورت کاراکتر نمایش داده می‌شود و در ابتدا همه خانه‌ها خالی می‌باشد چون با کاراکتر Space پر شده است.
- **String View:** در این حالت برعکس دو حالت قبل صفحه نمایش بصورت یک Editor می‌باشد و نه بصورت خانه‌های تفکیک شده و به راحتی می‌توان در این صفحه عبارات مورد نظر را تایپ کرد. در حالت‌های ۱ و ۲ باید ابتدا بر روی خانه مورد نظر با موس کلیک و سپس عدد مورد نظر و یا کاراکتر مورد نظر را وارد کرد. در صورتیکه عبارت Mouse Tracking چک مارک خورده باشد با حرکت موس خانه مورد نظر انتخاب و آماده ورود می‌شود.

علاوه بر این در آیکنهای بالای این بخش وجود دارد:

**Clear Data:** انتخاب این گزینه تمام اطلاعات موجود در صفحه اطلاعات را با کاراکتر Space جایگزین می‌نماید که در قسمت Decimal View همه خانه‌ها با عدد ۳۲ پر می‌شود.

**Fill Character:** تمام خانه‌های موجود را با یک کاراکتر یا عدد پر می‌کند بعد از انتخاب، پنجره‌ای جهت دریافت کاراکتر مورد نظر نمایان می‌شود بعد از ورود و تأیید تمام خانه‌های حافظه با همین کاراکتر پر خواهد شد



دقت داشته باشید در صورتیکه رشته وارد کنید فقط اولین کاراکتر در نظر گرفته می‌شود. بطور مثال در صورتیکه کاراکتر 'A' را وارد کنید تمام خانه‌ها در قسمت Decimal View مقدار ۶۵ و در قسمت Character View با کاراکتر 'A' پر می‌شوند. در صورتیکه بخواهید کد اسکی کاراکتر را وارد کنید باید قبل از ورود کد اسکی، کاراکتر # را وارد کنید.

در این صورت قسمت Decimal View با عدد وارد شده و قسمت Character View با کاراکتر مربوطه پر می‌شود بطور مثال در صورتیکه #65 را وارد کنید در قسمت اول عدد ۶۵ و در قسمت دو کاراکتر 'A' نمایش داده می‌شود. در هر حالت قسمت String View تمام کاراکترهای حافظه را بصورت رشته‌ای نمایش می‌دهد. Open Data: با استفاده از این گزینه می‌توان اطلاعات را از یک فایل متنی یا XML و یا NLD فراخوانی کرد. بعد از انتخاب پنجره‌ای جهت تعیین فایل نمایش داده می‌شود که پس از انتخاب در صورت معتبر بودن فایل، اطلاعات در قسمت‌های مختلف نمایش داده می‌شود.

لازم به توضیح است که فایل‌های NLD دارای رمز عبور می‌باشند. Save Data: اطلاعات وارد شده در بخش اطلاعات را می‌توان در یک فایل ذخیره نمود این فایل می‌تواند به یکی از صورت‌های متنی، XML و یا فرمت اختصاصی NLD ذخیره شود. Read From Device: در این قسمت می‌توانید اطلاعات موجود در شناسه را خوانده و در صفحه نمایش دهید. برای استفاده از این سرویس باید VID و Password وارد شده باشند. دقت داشته باشید که اطلاعات وارد شده در صفحه نمایش تا زمانی که فرمان ذخیره را صادر نکرده باشید در شناسه ذخیره نمی‌شوند. همچنین استفاده از این گزینه اطلاعات وارد شده در صفحه نمایش را با اطلاعات موجود در شناسه جایگزین می‌نماید.

Save Data in Device: این گزینه اطلاعات وارد شده در صفحه نمایش را در شناسه ذخیره می‌کند و اطلاعات قبلی موجود در شناسه حذف و اطلاعات جدید جایگزین می‌گردد. دقت داشته باشید این گزینه فقط بخش اطلاعات را ذخیره می‌کند.

View Mode: با توجه به انتخاب گزینه داخل این لیست جدول‌های بخش داده‌های غیر قابل دسترس را با رنگ قرمز نشانه‌گذاری می‌کند. علاوه بر این با استفاده از کلیدهای Current یا Model می‌توانید تعیین کنید که مبنای نشانه‌گذاری بر اساس تنظیمات سطح دسترسی شناسه (Current) موجود باشد یا بر اساس تنظیمات جاری در بخش Data Permissions (حالت Model) باشد.

در موقع استفاده از این بخش باید مقادیر VID و رمز عبور (Password) وارد شده باشند.



### ۳-۱-۶- تب تنظیمات سطوح دسترسی قسمت داده (Data Permissions)

Novin Lock Builder (C:\Documents and Settings\Fattahi\Desktop\1.NLD)

General Options | Timer/Counter | Data Section | **Data Permissions** | Net and Web options

Data Section Allocation Chart

| Segment | Size | Reader | Writer |
|---------|------|--------|--------|
| 1       | 16   | R:0    | W:0    |
| 2       | 32   | R:1    | W:1    |
| 3       | 64   | R:2    | W:2    |
| 4       | 128  | R:3    | W:3    |
| 5       | 32   | R:3    | W:0    |
| 6       | 128  | R:1    | W:1    |

Load Current Setting | Model Setting Below

| Partition   | Size | Reader             | Writer             | Offset |
|-------------|------|--------------------|--------------------|--------|
| Partition 1 | 16   | Everyone           | Everyone           | 0      |
| Partition 2 | 32   | Developer or Admin | Developer or Admin | 16     |
| Partition 3 | 64   | Just Administrator | Just Administrator | 48     |
| Partition 4 | 128  | Just End-User      | Just End-User      | 112    |
| Partition 5 | 32   | Just End-User      | Everyone           | 240    |
| Partition 6 | 128  | Developer or Admin | Developer or Admin | 272    |
|             |      |                    |                    | 400    |

Load Current Setting | Write New Setting

Device Ready( SN : 5.110.110.110 , MemorySize : 200 (Unicode))

با توجه به اینکه سطوح دسترسی در شناسه چهار سطح است، می‌توان از طریق این قسمت بخش داده شناسه را به حداکثر ۶ قسمت تقسیم کرد و برای هر قسمت سطح دسترسی خواندن و نوشتن مجزا تعریف کرد. توجه داشته باشید که این تنظیم برای ۵ قسمت اول انجام پذیر است. قسمت ششم همان باقیمانده فضای داده استفاده نشده در ۵ قسمت اول است که به صورت پیش فرض برای نوشتن یا خواندن آن سطح دسترسی دوم در نظر گرفته شده است. سایز قسمت‌ها نیز می‌توانید عددی از بین ۱۶، ۳۲، ۶۴، ۱۲۸، ۲۵۶ یا ۵۱۲ باشد. همچنین ابزار گرافیکی در این بخش شما را به درک بیشتر نحوه قسمت بندی بخش داده کمک می‌کند.



### ۳-۱-۷- تب تنظیمات شبکه و وب (Net and Web options)

The screenshot shows the 'Novin Lock Builder' application window. The 'Net and Web options' tab is active. It contains the following elements:

- General Options:** A section for 'Enter your NetLock Maximum user count' with radio buttons for 'None (Disable)', 'Special' (selected with a dropdown set to '6'), and 'Unlimited'. There are 'Get Net' and 'Set Net' buttons.
- Web Configuration:** A section with a checked 'Enable WebLock' checkbox. Below it, a status bar shows 'Total Remaining Available Web Config Space : 17 char'. There are input fields for 'Get Command Switch' (GetCMD), 'Kill Command Switch' (KillCMD), and 'Set Result Switch' (Result). There are also 'Get Web Config', 'Set Web Config', and 'Reset Web Config' buttons.
- Console URL:** An input field containing 'nl/console.asp'.
- WebSite URL:** An input field containing 'https://www.novinfazar.com'.
- Clear Form:** A button to reset the input fields.
- Template:** A section showing three URLs: 'Get URL', 'Kill URL', and 'Set URL', all pointing to 'https://www.novinfazar.com/nl/console.asp?Action=...'.
- Device Ready:** A green status bar at the bottom showing 'Device Ready( SN : 5.110.110.110 , MemorySize : 200 (Unicode))'.

در صورتی که می‌خواهید شناسه را به عنوان شناسه تحت شبکه استفاده کنید، لازم است پارامتر شبکه ای آن را فعال کنید. این پارامتر حاوی حداکثر کاربران (کامپیوترها) شبکه است که می‌توانند به صورت اشتراکی از شناسه استفاده کنند. این عدد می‌تواند بین ۱ تا ۲۵۵ باشد یا به صورت بینهایت تنظیم شده باشد. علاوه بر این در این تب شما می‌توانید مشخصات صفحه کنسول (Console) سرور وب خود را برای استفاده از شناسه در وب تنظیم نمایید. این پارامترها را در بخش وب این راهنما بیشتر توضیح داده شده است.

### ۳-۱-۸- برنامه ریزی شناسه

برای برنامه ریزی شناسه باید اعمال زیر انجام شود.

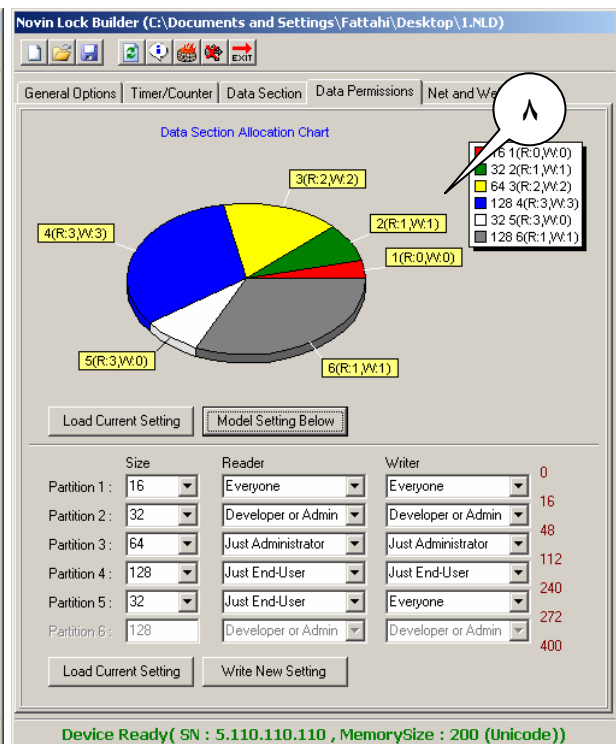
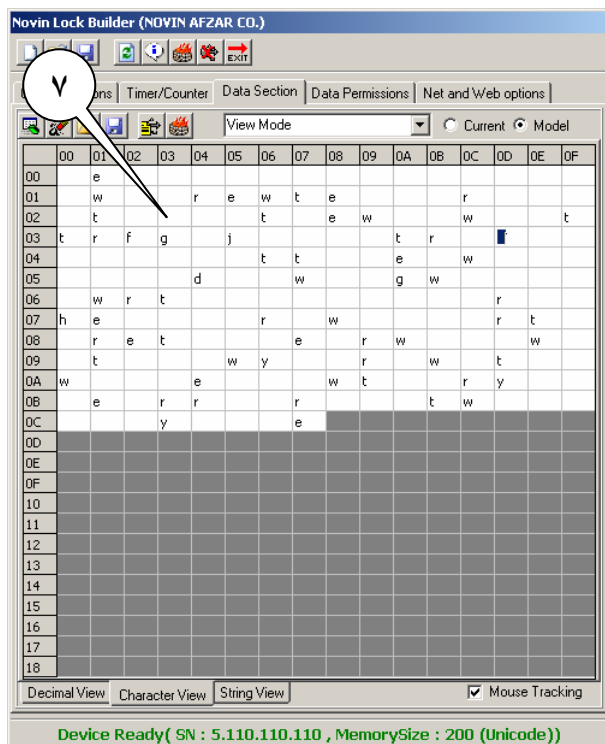
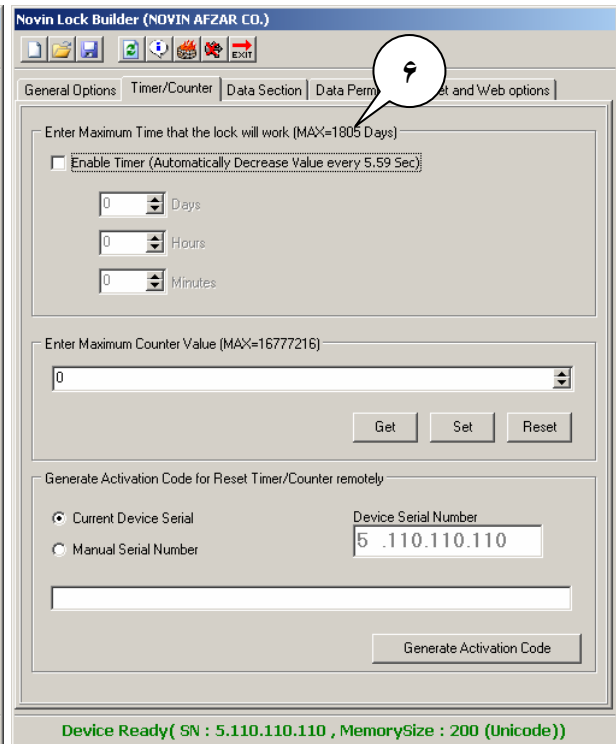
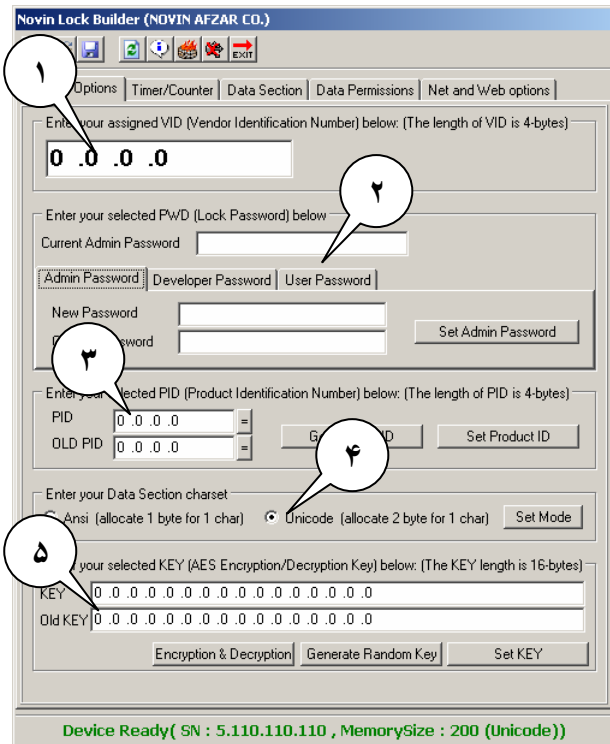
۱. ورود کد اختصاصی (VID)
۲. تنظیم کردن رمز های عبور (Administrator and Developer Passwords)
۳. ثبت کردن (PID)
۴. تنظیم charset فضای داده شناسه
۵. ست کردن کلید رمزنگاری همگانی
۶. تنظیمات مربوط به شناسه های استفاده محدود (Timer/Counter)
۷. ورود اطلاعاتی که باید در قسمت دیتای شناسه ذخیره شود.
۸. اعمال تقسیمات لازم برای قسمت داده و تعیین سطح دسترسی لازم برای خواندن یا نوشتن شناسه
۹. تنظیمات مربوط به شناسه شبکه

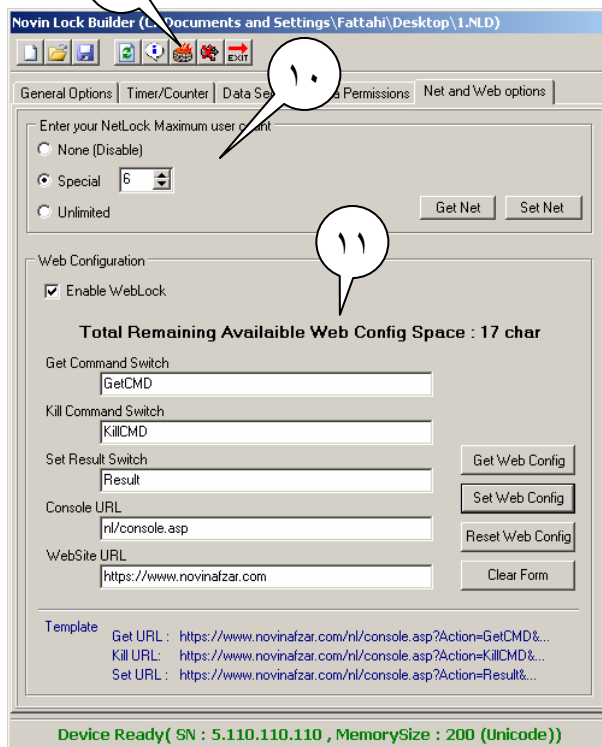


۱۰. مشخصات صفحه کنسول برای شناسه وب (در قسمت های مربوطه توضیح کامل دارد)

۱۱. نوشتن اطلاعات در شناسه

به موقعیت موارد بالا در شکل های زیر توجه نمایید.





### ۳-۱-۹- ایجاد شناسه های مشابه

برای این منظور از طریق ذخیره کردن تنظیمات در فایل آنرا برای استفاده های بعدی نگهداری کنید.



هنگامی که فایل را مجدداً باز می‌کنید در صورتی که به صورت NLD ذخیره کرده باشید، پنجره باز می‌شود که باید رمز عبور Administrator یا User آن فایل را وارد کنید. در صورتی که در مود مدیریت وارد شوید، علاوه بر نوشتن اطلاعات (مورد ۱۱ بند قبلی) می‌توانید محتوا و تنظیمات را تغییر دهید.

در صورتی که می‌خواهید عمل کپی سازی توسط اپراتور پایین دست انجام دهید، رمز عبور User را در اختیار وی قرار دهید، و از بابت نمایش داده نشدن پارامترهای تنظیم شده به او اطمینان داشته باشید. در صورتی که در مود User فایل باز شود پنجره شکل مقابل باز می‌شود که فقط امکان نوشتن اطلاعات در شناسه برای

کاربر امکانپذیر است. کاربر در این موقع پس از وارد کردن شناسه خام در پورت، پس از اطمینان از شناخته شدن صحیح شناسه (با توجه به مشخصات سبز رنگ در نوار پایین) کلید نوشتن را زده و پس از دریافت پیغام تایید پایان عملیات می‌تواند شناسه دیگری را به همین حالت برنامه ریزی کند.



### ۲-۳- ارتباط با شناسه در برنامه نویسی

متدهای استفاده از شناسه به سه نوع در دسترس برنامه نویسی قرار دارد. یک سری توابع که با `NL_` شروع شده و هر کدام مستقل از بقیه کار می‌کنند متدهایی که به صورت کلاسی از طریق `interface` های موجود روی `CD` قابل دسترس است. دو دسته بالا به صورت توابع قابل دسترس از بیرون `DLL` (`export`) پیاده شده است. دو کلاس اتومات مشتق شده از `NLLIB` به نامهای `clsNovinLock` و `clsNetClient` که در صورت ثبت (`register`) کردن `dll` در ویندوز قابل دسترس می‌شوند. برای فراخوانی شناسه از داخل نرم افزار باید فایل مربوط به روتین های شناسه (`NovinLock.dll`) را به پروژه اضافه شود. در برنامه نویسی `Borland` (`Delphi` و `CBuilder`) از هر سه دسته فوق می‌توان استفاده کرد. در برنامه نویسی `Microsoft` غیر `NET`. از دسته `A` به راحتی می‌توان استفاده کرد ولی در `NET`. دسته `C` بهتر و کارآمدتر است. برای درک بهتر می‌توانید نرم افزارهای نوشته شده در شاخه `Sample code` را بررسی نمائید.



## ۳-۲-۱- متدهای رابط سری A

روال های موجود در دسته A عبارتند از:

|                           |   |
|---------------------------|---|
| NL_SetDeviceType          | تعیین نوع شناسه                             |
| NL_GetDeviceReady         | تست وجود شناسه                              |
| NL_GetSerial              | دریافت شماره سریال شناسه                    |
| NL_GetVersion             | دریافت شماره نسخه شناسه                     |
| NL_GetMemorySize          | دریافت مقدار حافظه قابل دسترس               |
| NL_GetMemorySizeEx        | دریافت مقدار کاراکتر قابل دسترس             |
| NL_GetNET                 | دریافت تعداد حداکثر مجاز کاربران شناسه شبکه |
| NL_GetPID                 | دریافت PID                                  |
| NL_SetDataByte            | نوشتن یک بایت در حافظه شناسه                |
| NL_SetDataBlockStr        | نوشتن اطلاعات در حافظه بصورت رشته ای        |
| NL_GetDataByte            | خواندن یک بایت از حافظه شناسه               |
| NL_GetDataBlockStr        | خواندن اطلاعات از حافظه بصورت رشته ای       |
| NL_GetEncryption          | رمزنگاری                                    |
| NL_GetDecryption          | رمزگشایی                                    |
| NL_GetErrDescription      | تفسیر یک کد خطا به یک پیام                  |
| NL_ConvDNStringToString   | تبدیل رشته از DNString به رشته کاراکتری     |
| NL_ConvPWStringToDNString | تبدیل رشته از رشته کاراکتری به DNString     |

| NL_SetDeviceType   | تعیین نوع شناسه |
|--|-----------------|
| <b>NL_SetDeviceType(DeviceType:word) : WORD</b>  |                 |
| در ابتدای استفاده از روتین ها باید نوع شناسه تعیین گردد که در حال حاضر فقط یک مدل از شناسه ها عرضه شده است و باید از کد صفر استفاده کرد. |                 |
| این روتین باید شناسه از تمام روتین های دیگر برای یکبار اجرا شود .  |                 |
| <b>DeviceType</b> : کد مربوط به نوع شناسه می باشد که از نوع <b>Word</b> می باشد.   |                 |
| مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می شود.  |                 |



|   |                |
|---|----------------|
| NL_GetDeviceReady   | تست وجود شناسه |
| NL_GetDeviceReady() : Word  |                |
| ساده ترین روش برای چک کردن شناسه در پورت USB.   |                |
| هیچ پارامتری ندارد  |                |
| مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود. |                |

|  |                          |
|--|--------------------------|
| NL_GetSerial   | دریافت شماره سریال شناسه |
| NL_GetSerial(var sSerial:WideString):Word  |                          |
| هر شناسه دارای یک سریال یکتا می‌باشد که در موقع تولید، توسط تولید کننده در شناسه ذخیره می‌گردد با استفاده از این روتین می‌توان این سریال را بدست آورد. |                          |
| sSerial: بعد از اجرا، سریال در داخل این پارامتر ذخیره می‌گردد و نوع پارامتر WideString می‌باشد.  |                          |
| مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود.  |                          |

|  |                         |
|--|-------------------------|
| NL_GetVersion  | دریافت شماره نسخه شناسه |
| NL_GetVersion(var sVersion:WideString):Word  |                         |
| هر شناسه دارای نرم افزارهای داخلی می‌باشد که با توسعه شناسه ممکن است این نرم افزارها تغییر نماید. لذا هر شناسه دارای شماره نسخه مربوط به خود است که با استفاده از این روتین می‌توان شماره نسخه شناسه را بدست آورد. |                         |
| sVersion: بعد از اجرا ، شماره نسخه را در خود دارد و از نوع WideString می‌باشد.   |                         |
| مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود.  |                         |

|  |                               |
|--|-------------------------------|
| NL_GetMemorySize   | دریافت مقدار حافظه قابل دسترس |
| NL_GetMemorySize(var mem : WORD ) : WORD   |                               |
| با توجه به نوع شناسه، حافظه قابل دسترس متغیر می‌باشد، با استفاده از این سرویس، می‌توان مقدار این حافظه را به بایت بدست آورد. |                               |
| Mem : این پارامتر از نوع WORD می‌باشد و بعد از اجرا مقدار حافظه قابل دسترس را در خود نگه می‌دارد                             |                               |
| مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود.                                      |                               |



|   |                                 |
|---|---------------------------------|
| NL_GetMemorySizeEx  | دریافت مقدار کاراکتر قابل دسترس |
| NL_GetMemorySize(var mem : WideString) : WORD   |                                 |
| با توجه به نوع شناسه، حافظه قابل دسترس و charset فضای داده، متغیر می‌باشد، با استفاده از این سرویس، می‌توان مقدار این حافظه را برحسب کاراکتر و نوع چارست بدست آورد. |                                 |
| Mem: این پارامتر از نوع WideString هست و بعد از اجرا مقدار حافظه قابل دسترس را در خود نگه می‌دارد   |                                 |
| مقدار برگشتی: در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود.  |                                 |

|   |   |
|---|---|
| NL_GetNET   | دریافت حداکثر تعداد کاربران مجاز شناسه شبکه |
| NL_GetNET(var bNET:BYTE):word;  |   |
| با توجه تنظیمات شبکه این تابع تعداد حداکثر مجاز کاربران استفاده کننده از شناسه تحت شبکه را نشان می‌دهد. در صورت ست نشدن برای شبکه خطای ۶ (Code not set) برمی‌گرداند |   |
| bNET: این پارامتر از نوع BYTE می‌باشد و بعد از اجرا مقدار مورد نظر را در خود نگه می‌دارد  |   |
| مقدار برگشتی: در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود. در صورت ست نشدن برای شبکه خطای ۶ (Code not set) برمی‌گرداند                  |   |

|  |            |
|--|------------|
| NL_GetPID  | دریافت PID |
| NL_GetPID(sVID,sPWD:WideString; var sPID:WideString):Word  |            |
| PID ذخیره شده در شناسه توسط این سرویس قابل دسترسی می‌باشد. PID با استفاده از نرم افزار برنامه ریز در شناسه ذخیره می‌گردد |            |
| sVID: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت WideString باشد.                     |            |
| sPWD: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت WideString باشد.               |            |
| sPID: بعد از اجرای روتین، PID در این پارامتر قرار داده می‌شود.   |            |
| مقدار برگشتی: در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود.                                   |            |



| NL_SetDataByte   | نوشتن یک بایت در حافظه شناسه |
|--|------------------------------|
| <b>NL_SetDataByte(sVID , sPWD:WideString; Address:Word; DataByte:byte):Word</b>  |                              |
| <p>برای نوشتن اطلاعات در شناسه بصورت بایت به بایت از این روتین استفاده می‌شود این روتین در آدرس مشخص شده اطلاعات یک بایت را می‌نویسد.</p>  |                              |
| <p><b>sVID</b>: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت <b>WideString</b> باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت <b>WideString</b> باشد.</p> <p><b>Address</b>: آدرس مورد نظر جهت نوشتن اطلاعات و از نوع <b>Word</b> می‌باشد.</p> <p><b>DataByte</b>: داده مورد نظر در این پارامتر ذخیره و بعد از اجرای روتین ، در آدرس مشخص شده ذخیره می‌گردد. نوع آن <b>byte</b> می‌باشد.</p> |                              |
| <p>مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود.</p>   |                              |

| NL_SetDataBlockStr  | نوشتن اطلاعات در حافظه بصورت رشته ای |
|---|--------------------------------------|
| <b>NL_SetDataBlockStr(sVID,sPWD:WideString; StartAddress:word; DataLength:word; DataBlockStr:WideString; defChar:WideString = chr(32)):word</b>   |                                      |
| <p>در صورتیکه بخواهید بیش از یک بایت را بصورت رشته ای در حافظه ذخیره نمایید از این روتین استفاده کنید.</p>  |                                      |
| <p><b>sVID</b>: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت <b>WideString</b> باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت <b>WideString</b> باشد.</p> <p><b>StartAddress</b> : آدرس شروع برای ذخیره اطلاعات و از نوع <b>Word</b></p> <p><b>DataLength</b>: این متغیر که از نوع <b>Word</b> می‌باشد اندازه بلوک مورد نظر برای نوشتن را مشخص می‌نماید.</p> <p><b>DataBlockStr</b>: اطلاعات مورد نظر در این متغیر که یک آرایه از نوع <b>WideString</b> می‌باشد ذخیره و پس از اجرای روتین در شناسه نوشته می‌شود.</p> <p><b>defChar</b>: در صورتیکه رشته مورد نظر کمتر از طول ذکر شده باشد بقیه حافظه با این کاراکتر پر می‌شود .</p> |                                      |
| <p>مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود.</p>  |                                      |



| NL_GetDataByte   | خواندن یک بایت از حافظه شناسه |
|--|-------------------------------|
| <b>NL_GetDataByte(sVID , sPWD:WideString; Address:Word; var DataByte:byte):Word</b>  |                               |
| <p>برای خواندن اطلاعات از شناسه بصورت بایت به بایت از این روتین استفاده می‌شود این روتین از آدرس مشخص شده اطلاعات یک بایت را می‌خواند.</p>   |                               |
| <p><b>sVID</b>: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت <b>WideString</b> باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت <b>WideString</b> می‌باشد.</p> <p><b>Address</b>: آدرس مورد نظر جهت خواندن اطلاعات و از نوع <b>Word</b> می‌باشد.</p> <p><b>DataByte</b>: بعد از اجرا مقدار موجود در آدرس مورد نظر در این پارامتر ذخیره می‌گردد و نوع آن <b>byte</b> می‌باشد.</p> |                               |
| <p>مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود.</p>   |                               |

| NL_GetDataBlockStr   | خواندن اطلاعات از حافظه بصورت رشته ای |
|--|---------------------------------------|
| <b>NL_GetDataBlockStr(sVID , sPWD:WideString; StartAddress:Word; DataLength:Word; var DataBlockStr:WideString ):Word</b>   |                                       |
| <p>در صورتیکه بخواهید بیش از یک بایت را بصورت رشته ای از حافظه بخوانید باید از این روتین استفاده نمایید.</p>   |                                       |
| <p><b>sVID</b>: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت <b>WideString</b> باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت <b>WideString</b> باشد.</p> <p><b>StartAddress</b>: آدرس شروع برای خواندن اطلاعات و از نوع <b>Word</b></p> <p><b>DataLength</b>: این متغیر که از نوع <b>Word</b> می‌باشد اندازه بلوک مورد نظر برای خواندن را مشخص می‌نماید.</p> <p><b>DataBlockStr</b>: اطلاعات مورد نظر پس از اجرای روتین در این متغیر که یک آرایه از نوع <b>WideString</b> می‌باشد ذخیره می‌شود.</p> |                                       |
| <p>مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می‌شود.</p>   |                                       |



| NL_GetEncryption   | رمزنگاری |
|--|----------|
| <b>NL_GetEncryption(sVID , sPWD : WideString;sPData:WideString; Repetition:Word;<br/>var sCData:WideString):Word</b>   |          |
| <p>یکی از روتین های مهم و کاربردی در شناسه عمل رمزنگاری ( Encryption ) می باشد. این روتین با استفاده از کلید ذخیره شده در خود شناسه ، اطلاعات ارسالی را رمزنگاری می نماید ، این عمل می تواند بصورت متوالی انجام شود به این صورت که پارامتر ارسالی بعد از رمزنگاری شدن، دوباره تکرار شود.</p>   |          |
| <p><b>sVID</b>: این پارامتر VID می باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت WideString باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت WideString باشد.</p> <p><b>sPData</b> اطلاعات مورد نظر برای رمزنگاری بصورت WideString در این پارامتر قرار می گیرد.</p> <p><b>Repetition</b>: این متغیر که از نوع Word می باشد تعداد دفعات برای انجام عمل رمزنگاری را مشخص می کند این عدد حداقل باید یک باشد.</p> <p><b>sCData</b>: اطلاعات رمزنگاری شده بعد از اجرای متد در این پارامتر ذخیره می گردد که نوع WideString دارد.</p> |          |
| <p>مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می شود.</p>   |          |

| NL_GetDecryption   | رمزگشایی |
|--|----------|
| <b>NL_GetDecryption(sVID , sPWD : WideString;sCData:WideString; Repetition:Word;<br/>var sPData:WideString):Word</b>   |          |
| <p>برای رمزگشایی اطلاعات رمزنگاری شده از این روتین استفاده می شود. به این صورت که در صورتیکه عددی را رمزنگاری و سپس عدد بدست آمده را رمزگشایی نمائیم ، عدد اول بدست می آید به این شرط که تعداد تکرار در هر دو مرحله یکسان باشد.</p>  |          |
| <p><b>sVID</b>: این پارامتر VID می باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت WideString باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت WideString باشد.</p> <p><b>sCData</b>: اطلاعات رمزنگاری شده که می خواهیم رمزگشایی شود و به صورت WideString می باشد.</p> <p><b>Repetition</b>: این متغیر که از نوع Word می باشد تعداد دفعات برای انجام عمل رمزگشایی را مشخص می کند.</p> <p><b>sPData</b>: اطلاعات رمزگشایی شده بعد از اجرای متد در این پارامتر ذخیره می گردد که نوع WideString دارد.</p> |          |
| <p>مقدار برگشتی : در صورت اجرای موفقیت آمیز صفر و در غیر اینصورت کد خطا برگشت داده می شود.</p>   |          |



|  |                            |
|--|----------------------------|
| NL_GetErrDescription   | تفسیر یک کد خطا به یک پیام |
| NL_GetErrDescription(Err_NO:Word ;var Err_Desc:WideString);  |                            |
| کد خطای بازگشتی از توابع دیگر هر کدام معنی خاصی دارد که با این تابع می‌توانیم به آن دسترسی پیدا کرد. |                            |
| Err_NO: این پارامتر به عنوان ورودی به تابع داده می‌شود که همان کد خطای مورد نظر است                  |                            |
| Err_Desc: پیام مرتبط با کد خطا در این پارامتر ذخیره شده و بازگشت داده می‌شود                         |                            |
| مقدار برگشتی : خروجی خاصی ندارد  |                            |

|   |   |
|---|---|
| NL_ConvDNStringToString   | تبدیل رشته از DNString به رشته کاراکتری |
| NL_ConvDNStringToString(DNStr:WideString; var PWStr:WideString):boolean   |   |
| از رشته‌ها به دو صورت می‌توان استفاده کرد<br>به صورت معمولی ( کاراکتری ) و از نوع WideString<br>به صورت اعداد جدا شده با نقطه مانند آدرس IP<br>با استفاده از این روتین می‌توان رشته کاراکتری را به DN تبدیل کرد |   |
| DNStr: این پارامتر رشته ورودی می‌باشد که می‌خواهیم تبدیل کنیم و به صورت WideString می‌باشد.   |   |
| sPWD: بعد از اجرا رشته تبدیل شده در این پارامتر ذخیره می‌شود و بصورت کاراکتری و WideString باشد.  |   |
| مقدار برگشتی : در صورت اجرای موفقیت آمیز True و در غیر اینصورت False برگشت داده می‌شود.   |   |

|   |   |
|---|---|
| NL_ConvPWStringToDNString   | تبدیل رشته از رشته کاراکتری به DNString |
| NL_ConvPWStringToDNString(PWStr:WideString; var DNStr:WideString):boolean   |   |
| از رشته‌ها به دو صورت می‌توان استفاده کرد<br>به صورت معمولی ( کاراکتری ) و از نوع WideString<br>به صورت اعداد جدا شده با نقطه مانند آدرس IP<br>با استفاده از این روتین می‌توان رشته کاراکتری را به DN تبدیل کرد |   |
| sPWD: این پارامتر رشته ورودی می‌باشد که می‌خواهیم تبدیل کنیم و بصورت کاراکتری و WideString باشد.  |   |
| DNStr: بعد از اجرا رشته تبدیل شده در این پارامتر ذخیره می‌شود و به صورت WideString می‌باشد.   |   |
| مقدار برگشتی : در صورت اجرای موفقیت آمیز True و در غیر اینصورت False برگشت داده می‌شود.   |   |



## ۲-۲-۳- متدهای رابط سری B

روال های موجود در دسته B که در TclsNovinLock موجودند عبارتند از:

## Methods:

|                             |   |
|-----------------------------|---|
| SetDeviceType               | تعیین نوع شناسه                             |
| GetDeviceReady              | تست وجود شناسه                              |
| GetDeviceStatus             | تست وضعیت شناسه                             |
| GetSerial                   | دریافت شماره سریال شناسه                    |
| GetVersion                  | دریافت شماره نسخه شناسه                     |
| GetMemorySize               | دریافت مقدار حافظه قابل دسترس               |
| GetMemorySizeEx             | دریافت مقدار کاراکتر قابل دسترس             |
| GetNET                      | دریافت تعداد حداکثر مجاز کاربران شناسه شبکه |
| GetPID                      | دریافت PID                                  |
| SetDataByte                 | نوشتن یک بایت در حافظه شناسه                |
| SetDataBlockStr             | نوشتن اطلاعات در حافظه بصورت رشته ای        |
| GetDataByte                 | خواندن یک بایت از حافظه شناسه               |
| GetDataBlockStr             | خواندن اطلاعات از حافظه بصورت رشته ای       |
| GetEncryption               | رمزنگاری                                    |
| GetDecryption               | رمزگشایی                                    |
| ConvDelimitedStringToString | تبدیل رشته از DNString به رشته کاراکتری     |
| ConvStringToDelimitedString | تبدیل رشته از رشته کاراکتری به DNString     |
| SetUserPWD                  | تنظیم و تغییر سطح سوم دسترسی                |
| SetUserKEY                  | تنظیم کلید رمزنگاری شخصی (کاربر)            |
| GetTimerCounter             | دریافت مقدار شمارنده/زمانسنج                |
| Activate                    | فعال سازی مجدد قفل                          |
| DecreaseCounter             | کاهش یک واحد از شمارنده                     |
| Properties:                 |   |
| rrNo                        | کد خطای حاصل از آخرین متد اجرا شده          |
| ErrDescr                    | پیام خطای متناسب با کد خطا فوق              |
| UserPwd                     | رمز عبور سطح سه (کاربر)                     |



| SetDeviceType  | تعیین نوع شناسه |
|--|-----------------|
| procedure SetDeviceType(devType: Word);  |                 |
| در ابتدای استفاده از روتین ها باید نوع شناسه تعیین گردد که در حال حاضر فقط یک مدل از شناسه ها عرضه شده است و باید از کد صفر استفاده کرد. |                 |
| این روتین باید شناسه از تمام روتین های دیگر برای یکبار اجرا شود .  |                 |
| devType: کد مربوط به نوع شناسه می باشد که از نوع Word می باشد.   |                 |

| GetDeviceReady                                | تست وجود شناسه |
|---|----------------|
| procedure GetDeviceReady;                     |                |
| ساده ترین روش برای چک کردن شناسه در پورت USB. |                |
| هیچ پارامتری ندارد                            |                |
| هیچ مقدار بازگشتی ندارد                       |                |

| GetDeviceStatus   | تست وضعیت شناسه |
|---|-----------------|
| procedure GetDeviceStatus;  |                 |
| تست وضعیت: فعال یا یکی از وضعیتهای غیرفعال (Suspend, Disabled or Catastrophic Disabled) |                 |
| هیچ پارامتری ندارد  |                 |
| هیچ مقدار بازگشتی ندارد. وضعیت موجود قفل در ErrNo با خطاهای متناسب برگشت می شود.        |                 |

| GetSerial  | دریافت شماره سریال شناسه |
|--|--------------------------|
| function GetSerial: WideString;  |                          |
| هر شناسه دارای یک سریال یکتا می باشد که در موقع تولید، توسط تولید کننده در شناسه ذخیره می گردد با استفاده از این روتین می توان این سریال را بدست آورد. |                          |
| هیچ پارامتری ندارد   |                          |
| مقدار برگشتی : در صورت اجرای موفقیت آمیز شماره سریال را که از نوع WideString می باشد باز میگرداند.   |                          |



|   |                         |
|---|-------------------------|
| GetVersion  | دریافت شماره نسخه شناسه |
| function GetVersion: WideString;  |                         |
| هر شناسه دارای نرم افزارهای داخلی می باشد که با توسعه شناسه این نرم افزارها تغییر می کند. لذا هر شناسه دارای شماره نسخه مربوط به خود است که از طریق این روتین می توان آنرا بدست آورد. |                         |
| هیچ پارامتری ندارد  |                         |
| مقدار برگشتی : در صورت اجرای موفقیت آمیز شماره نسخه از نوع WideString برگشت داده می شود.  |                         |

|  |                               |
|--|-------------------------------|
| GetMemorySize  | دریافت مقدار حافظه قابل دسترس |
| function GetMemorySize: Word;  |                               |
| با توجه به نوع شناسه، حافظه قابل دسترس متغیر می باشد، با استفاده از این سرویس، می توان مقدار این حافظه را به بایت بدست آورد. |                               |
| هیچ پارامتری ندارد   |                               |
| مقدار برگشتی : از نوع WORD می باشد و بعد از اجرا مقدار حافظه قابل دسترس را برگشت می دهد.                                     |                               |

|   |                                 |
|---|---------------------------------|
| GetMemorySizeEx   | دریافت مقدار کاراکتر قابل دسترس |
| function GetMemorySizeEx: WideString;   |                                 |
| با توجه به نوع شناسه، حافظه قابل دسترس و charset فضای داده، متغیر می باشد، با استفاده از این سرویس، می توان مقدار این حافظه را برحسب کاراکتر و نوع چارست بدست آورد. |                                 |
| هیچ پارامتری ندارد  |                                 |
| مقدار برگشتی : از نوع Widestring می باشد، تعداد کاراکترهای حافظه و نوع charset را برگشت می دهد.   |                                 |

|   |   |
|---|---|
| GetNET  | دریافت حداکثر تعداد کاربران مجاز شناسه شبکه |
| function GetNET: Word;  |   |
| با توجه تنظیمات شبکه این تابع تعداد حداکثر مجاز کاربران استفاده کننده از شناسه تحت شبکه را نشان می دهد. در صورت ست نشدن برای شبکه خطای (Code not set) برمی گرداند |   |
| هیچ پارامتری ندارد  |   |
| مقدار برگشتی : از نوع BYTE می باشد و بعد از اجرا مقدار مورد نظر را بر می گرداند. در صورت ست نشدن برای شبکه خطای (Code not set) اتفاق می افتد                      |   |



| GetPID  | دریافت PID |
|---|------------|
| <pre>function GetPid(const sVID: WideString; const sPWD: WideString): WideString;</pre>   |            |
| <p>PID ذخیره شده در شناسه توسط این سرویس قابل دسترسی می‌باشد. PID با استفاده از نرم افزار برنامه ریز در شناسه ذخیره می‌گردد</p>   |            |
| <p>sVID: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت WideString باشد.</p> <p>sPWD: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت WideString باشد.</p> |            |
| <p>مقدار برگشتی: در صورت اجرای موفقیت آمیز مقدار PID به صورت Widestring برگشت داده می‌شود.</p>  |            |

| SetDataByte   | نوشتن یک بایت در حافظه شناسه |
|---|------------------------------|
| <pre>procedure SetDataByte(const sVID: WideString; const sPWD: WideString; Address: Word;DataByte: Byte);</pre>   |                              |
| <p>برای نوشتن اطلاعات در شناسه بصورت بایت به بایت از این روتین استفاده می‌شود این روتین در آدرس مشخص شده اطلاعات یک بایت را می‌نویسد.</p>   |                              |
| <p>sVID: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت WideString باشد.</p> <p>sPWD: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت WideString باشد.</p> <p>Address: آدرس مورد نظر جهت نوشتن اطلاعات و از نوع Word می‌باشد.</p> <p>DataByte: داده مورد نظر در این پارامتر ذخیره و بعد از اجرای روتین، در آدرس مشخص شده ذخیره می‌گردد. نوع آن byte می‌باشد.</p> |                              |
| <p>هیچ مقدار بازگشتی ندارد</p>  |                              |



| SetDataBlockStr   | نوشتن اطلاعات در حافظه بصورت رشته ای |
|---|--------------------------------------|
| <pre>procedure SetDataBlockStr(const sVID: WideString; const sPWD: WideString; Start: Word; DataLen: Word; const DataBlockStr: WideString; const defChar: WideString= ' ');</pre>   |                                      |
| <p>در صورتیکه بخواهید بیش از یک بایت را بصورت رشته ای در حافظه ذخیره نمایید باید از این روتین استفاده نمایید.</p>   |                                      |
| <p><b>sVID</b>: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت WideString باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت WideString باشد.</p> <p><b>Start</b>: آدرس شروع برای ذخیره اطلاعات و از نوع Word</p> <p><b>DataLen</b>: این متغیر که از نوع Word می‌باشد اندازه بلوک مورد نظر برای نوشتن را مشخص می‌نماید.</p> <p><b>DataBlockStr</b>: اطلاعات مورد نظر در این متغیر که یک آرایه از نوع WideString می‌باشد ذخیره و پس از اجرای روتین در شناسه نوشته می‌شود.</p> <p><b>defChar</b>: در صورتیکه رشته مورد نظر کمتر از طول ذکر شده باشد بقیه حافظه با این کاراکتر پر می‌شود .</p> |                                      |
| <p>هیچ مقدار بازگشتی ندارد</p>  |                                      |

| GetDataByte   | خواندن یک بایت از حافظه شناسه |
|---|-------------------------------|
| <pre>function GetDataByte(const sVID: WideString; const sPWD: WideString; Address: Word): Byte;</pre>   |                               |
| <p>برای خواندن اطلاعات از شناسه بصورت بایت به بایت از این روتین استفاده می‌شود این روتین از آدرس مشخص شده اطلاعات یک بایت را می‌خواند.</p>  |                               |
| <p><b>sVID</b>: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت WideString باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت WideString می‌باشد.</p> <p><b>Address</b>: آدرس مورد نظر جهت خواندن اطلاعات و از نوع Word می‌باشد.</p> |                               |
| <p>مقدار برگشتی: در صورت اجرای موفقیت مقدار موجود در آدرس مورد نظر برگشت داده می‌شود و نوع آن byte می‌باشد.</p>   |                               |



| GetDataBlockStr   | خواندن اطلاعات از حافظه بصورت رشته ای |
|---|---------------------------------------|
| <pre>function GetDataBlockStr(const sVID: WideString; const sPWD: WideString; Start: Word; DataLen: Word): WideString;</pre>  |                                       |
| <p>در صورتیکه بخواهید بیش از یک بایت را بصورت رشته ای از حافظه بخوانید باید از این روتین استفاده نمایید.</p>  |                                       |
| <p><b>sVID</b>: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت WideString باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت WideString باشد.</p> <p><b>Start</b>: آدرس شروع برای خواندن اطلاعات و از نوع Word</p> <p><b>DataLen</b>: این متغیر که از نوع Word می‌باشد اندازه بلوک مورد نظر برای خواندن را مشخص می‌نماید.</p> |                                       |
| <p>مقدار برگشتی : اطلاعات مورد نظر پس از اجرای موفقیت آمیز روتین بصورت یک widestring برگشت داده می‌شود.</p>   |                                       |

| GetEncryption   | رمزنگاری |
|---|----------|
| <pre>function GetEncryption(const sVID: WideString; const sPWD: WideString; const sPData: WideString; Repetition: Word): WideString;</pre>  |          |
| <p>یکی از روتین های مهم و کاربردی در شناسه عمل رمزنگاری (Encryption) می‌باشد.</p> <p>این روتین با استفاده از کلید ذخیره شده در خود شناسه ، اطلاعات ارسالی را رمزنگاری می‌نماید ، این عمل می‌تواند بصورت متوالی انجام شود به این صورت که پارامتر ارسالی بعد از رمزنگاری شدن ، دوباره رمزنگاری شود.</p>   |          |
| <p><b>sVID</b>: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت WideString باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت WideString باشد.</p> <p><b>sPData</b> اطلاعات مورد نظر برای رمزنگاری بصورت WideString در این پارامتر قرار می‌گیرد.</p> <p><b>Repetition</b>: این متغیر که از نوع Word می‌باشد تعداد دفعات برای انجام عمل رمزنگاری را مشخص می‌کند این عدد حداقل باید یک باشد.</p> |          |
| <p>مقدار برگشتی : اطلاعات رمزنگاری شده بعد از اجرای موفقیت آمیز روتین و از نوع WideString برگشت داده می‌شود.</p>  |          |



| GetDecryption   | رمزگشایی |
|---|----------|
| <pre>function GetDecryption(const sVID: WideString; const sPWD: WideString; const sCData: WideString; Repetition: Word): WideString;</pre>  |          |
| <p>برای رمزگشایی اطلاعات رمزنگاری شده از این روتین استفاده می‌شود. به این صورت که در صورتیکه عددی را رمزنگاری و سپس عدد بدست آمده را رمزگشایی نمائیم ، عدد اول بدست می‌آید به این شرط که تعداد تکرار در هر دو مرحله یکسان باشد.</p>   |          |
| <p><b>sVID</b>: این پارامتر VID می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و باید بصورت WideString باشد.</p> <p><b>sPWD</b>: این پارامتر رمز عبور شناسه می‌باشد که قبل از اجرای روتین باید مقدار دهی شود و بصورت WideString باشد.</p> <p><b>sCData</b>: اطلاعات رمزنگاری شده که می‌خواهیم رمزگشایی شود و به صورت WideString می‌باشد.</p> <p><b>Repetition</b>: این متغیر که از نوع Word می‌باشد تعداد دفعات برای انجام عمل رمزگشایی را مشخص می‌کند.</p> |          |
| <p>مقدار برگشتی : در صورت اجرای موفقیت آمیز : اطلاعات رمزگشایی شده برگشت داده می‌شود که از نوع WideString می‌باشد.</p>  |          |

| ConvDelimitedStringToString  | تبدیل رشته از DNString به رشته کاراکتری |
|--|---|
| <pre>function ConvDelimitedStringToString(const strDCnv: WideString;const ALen: Word=16;const Sep: WideString='.'): WideString;</pre>  |   |
| <p>از رشته ها به دو صورت می‌توان استفاده کرد به صورت معمولی ( کاراکتری ) و از نوع WideString به صورت اعداد جدا شده با نقطه مانند آدرس IP با استفاده از این روتین می‌توان رشته کاراکتری را به DN تبدیل کرد</p>  |   |
| <p><b>strDCnv</b>: این پارامتر رشته ورودی می‌باشد که می‌خواهیم تبدیل کنیم و به صورت WideString می‌باشد.</p> <p><b>ALen</b>: طول رشته ای که قرار است تبدیل شود را مشخص می‌کند.</p> <p><b>Sep</b>: کاراکتر واسط در خروجی را تعیین می‌کند که به صورت پیشفرض '.' است</p> |   |
| <p>مقدار برگشتی : در صورت اجرای موفقیت آمیز رشته تبدیل شده بصورت کاراکتری و WideString برگشت داده می‌شود.</p>  |   |



| ConvStringToDelimitedString   | تبدیل رشته از رشته کاراکتری به DNString |
|---|---|
| <pre>function ConvStringToDelimitedString(const strCnv: WideString;const ALen: Word=16;const Sep: WideString='.'): WideString;</pre>  |   |
| <p>از رشته ها به دو صورت می توان استفاده کرد<br/>به صورت معمولی ( کاراکتری ) و از نوع WideString<br/>به صورت اعداد جدا شده با نقطه مانند آدرس IP<br/>با استفاده از این روتین می توان رشته کاراکتری را به DN تبدیل کرد</p> |   |
| <p>strCnv: این پارامتر رشته ورودی می باشد که می خواهیم تبدیل کنیم و رشته ای از نوع WideString باشد.<br/>Sep: کاراکتر واسط در خروجی را تعیین می کند که به صورت پیش فرض '.' است</p>   |   |
| <p>مقدار برگشتی : در صورت اجرای موفقیت آمیز رشته تبدیل شده بصورت کاراکتری و WideString برگشت داده می شود.</p>   |   |

| SetUserPWD   | تنظیم و تغییر سطح سوم دسترسی |
|--|------------------------------|
| <pre>Procedure SetUserPWD(sOUPWD,sUPWD:WideString);</pre>  |                              |
| <p>رمز عبور سطح سوم (کاربر) توسط این روتین قابل تنظیم و تغییر است.</p>   |                              |
| <p>sOUPWD: رمز عبور فعلی که برای اولین بار خالی است.<br/>sUPWD: رمز عبور جدید که قرار است تنظیم شود.<br/>هر دو پارامتر بالا از نوع widestring است و باید به فرمت DNString به تابع داده شود</p> |                              |
| <p>هیچ مقدار بازگشتی ندارد</p>   |                              |

| SetUserKEY   | تنظیم کلید رمزنگاری شخصی (کاربر) |
|--|----------------------------------|
| <pre>Procedure SetUserKEY(sUPWD,sNKEY:WideString);</pre>   |                                  |
| <p>برای انجام رمزنگاری شخصی یک کلید ۱۶ بیتی (۱۲۸ بیتی) در شناسه تنظیم می شود<br/>توجه داشته باشید این کلید فقط می تواند یک بار در شناسه ست شود و هنگام تنظیم مجدد خطا خواهد داد.</p> |                                  |
| <p>sUPWD: رمز عبور فعلی کاربر (سطح سوم)<br/>sNKEY: کلیدی که قرار است تنظیم شود.<br/>هر دو پارامتر بالا از نوع widestring است و باید به فرمت DNString به تابع داده شود</p>            |                                  |
| <p>هیچ مقدار بازگشتی ندارد</p>   |                                  |



|   |                             |
|---|-----------------------------|
| GetTimerCounter   | دریافت مقدار شمارنده/زمانسج |
| <b>Function GetTimerCounter: integer;</b>   |                             |
| در صورتی که تنظیمات خاصی برای محدود کردن دفعات یا مدت زمان استفاده از قفل تعیین شده باشد با استفاده از این روتین می‌توان از آن تنظیمات مطلع شد.             |                             |
| هیچ پارامتری ندارد  |                             |
| مقدار بازگشتی مقدار عدد موجود در پارامتر شمارنده قفل است. در صورتی که کاهنده خودکار (Timer) ست نشده باشد خطای Nlevt_TimerNS در پارامتر خطا مشاهده خواهد شد. |                             |

|   |                   |
|---|-------------------|
| Activate  | فعالسازی مجدد قفل |
| <b>procedure Activate(ActivationCode: WideString);</b>  |                   |
| در صورتی که در طول استفاده از شناسه مقدار شمارنده به صفر برسد، شناسه به حالت Suspend می‌رود که برای خروج از این حالت و غیر فعال کردن شمارنده/زمانسج می‌توان با استفاده از یک کد فعال سازی صحیح شناسه را از طریق این روتین فعال کرد. |                   |
| ActivationCode: رشته فعالسازی است که توسط NLBuilder ایجاد می‌گردد.  |                   |
| هیچ مقدار بازگشتی ندارد   |                   |

|  |                         |
|--|-------------------------|
| DecreaseCounter  | کاهش یک واحد از شمارنده |
| <b>procedure DecreaseCounter;</b>  |                         |
| در طول استفاده از شناسه می‌توان مقداری را به عنوان دفعات استفاده از شناسه تعیین و با استفاده از این روتین در مقاطع مورد نظر آنرا یک واحد کاهش داد. |                         |
| هیچ پارامتری ندارد   |                         |
| هیچ مقدار بازگشتی ندارد  |                         |



| ErrNo   | کد خطای حاصل از آخرین متد اجرا شده |
|---|------------------------------------|
| <b>property ErrNo: Word</b>   |                                    |
| توابع فوق در صورت اجرای موفق باعث بازگشت مقدار - - از طریق این <b>property</b> می‌شوند ولی در صورت بروز خطا این پارامتر عدد غیر صفر بر می‌گرداند. |                                    |
| این مشخصه به طور یکطرفه خواندنی است و مقدار عددی از نوع <b>word</b> بر می‌گرداند  |                                    |

| ErrDescr   | پیام خطای متناسب با کد خطا فوق |
|--|--------------------------------|
| <b>property ErrDescr: WideString</b>   |                                |
| توابع در صورت اجرای موفق باعث بازگشت کلمه 'OK' از طریق این <b>property</b> می‌شوند ولی در صورت بروز خطا پیام متناسب با نوع خطا از این مشخصه قابل دریافت است. |                                |
| این مشخصه به طور یکطرفه خواندنی است و مقدار عددی از نوع <b>widestring</b> بر می‌گرداند   |                                |

### ۳-۲-۳- متدهای رابط سری C

در دسته C دو کلاس وجود دارد که روتینهای آن کاملاً مشابه دسته B عمل می‌کنند. ولی در برخی موارد دیگری اضافی دارند که به شرح زیر است :

کلاس NLLIB.clsNovinLock علاوه بر متدهای دسته B یک متد دیگر دارد که در استفاده از شناسه به عنوان شناسه وب مورد استفاده قرار می‌گیرد:

| GetWebCMD  | اجرای یک سری دستورالعمل از روی وب سرور |
|--|--|
| <b>function GetWebCMD(const ReferID: WideString): WideString;</b>  |  |
| هنگام ایجاد یک سری دستورالعمل کد رفرنس آن از طریق این متد به شناسه از اطلاع داده می‌شود                              |  |
| <b>ReferID</b> : آدرس رفرنس دستورالعملی است که روی وب سرور قرار دارد و از طریق صفحه کنسول به کلاس تحویل داده می‌شود. |  |
| مقدار برگشتی : در صورت اجرای موفق این متد ، رفرنس مربوط به نتایج برمیگردد تا بتوان در مراحل بعدی از آن استفاده کرد.  |  |



## ۳-۲-۴- متدهای رابط در شبکه

کلاس NLLIB.clsNetClient برای برقراری ارتباط و نحوه آن با NetServer متدهای زیر را در اختیار ما قرار می‌دهد. همچنین متدهای دسته B نیز در اینجا قابل دسترس است با این تفاوت که نتیجه متدها از شناسه روی شبکه که به طور اشتراکی استفاده می‌شود اخذ می‌شود.

## Methods:

|       |                     |
|-------|---------------------|
| Start | شروع ارتباط با سرور |
| Stop  | قطع ارتباط با سرور  |

## Properties:

|      |                          |
|------|--------------------------|
| Host | آدرس IP سرور             |
| Port | پورت سرویس دهی Netserver |

|  |                     |
|--|---------------------|
| Start  | شروع ارتباط با سرور |
| <b>procedure Start;</b>  |                     |
| پس از تعیین آدرس و پورت سرور این متد ارتباط به سرور مورد نظر را برقرار می‌کند. |                     |
| هیچ پارامتری ندارد   |                     |
| هیچ مقدار بازگشتی ندارد  |                     |

|   |                    |
|---|--------------------|
| Stop                                    | قطع ارتباط با سرور |
| <b>procedure Stop;</b>                  |                    |
| ارتباط مورد نظر با سرور را قطع می‌نماید |                    |
| هیچ پارامتری ندارد                      |                    |
| هیچ مقدار بازگشتی ندارد                 |                    |

|   |              |
|---|--------------|
| Host  | آدرس IP سرور |
| <b>property Host: WideString</b>                                |              |
| تعیین می‌کند سرور در کدام آدرس شبکه قرار دارد مثلاً 192.168.0.1 |              |
| این مشخصه به صورت خواندنی و نوشتنی عمل می‌کند                   |              |



| Port   | پورت سرویس دهی Netserver |
|--|--------------------------|
| property Port: Word                                      |                          |
| تعیین می‌کند کدام پورت سرور برای سرویسدهی تعیین شده است. |                          |
| این مشخصه به صورت خواندنی و نوشتنی عمل می‌کند            |                          |

### ۲-۲-۵- انواع خطاهای بازگشتی

در جدول زیر تمام خطاهای ممکن در استفاده از کتابخانه رابط شناسه در برنامه نویسی آمده است:

| شماره خطا | سمبل خطا  | توضیح                       |
|-----------|-----------|-----------------------------|
| ۰         | NLevt_OK  | بدون خطا                    |
| ۱         | NLevt_DNF | شناسه پیدا نشد              |
| ۲         | NLevt_INP | پارامتر ورودی غیرمعتبر است  |
| ۳         | NLevt_IOE | خطای IO                     |
| ۴         | NLevt_WRP | پارامتر ورودی اشتباه است    |
| ۵         | NLevt_CS  | کد مورد نظر ست شده است      |
| ۶         | NLevt_CNS | کد مورد نظر ست نشده است     |
| ۸         | NLevt_BSY | شناسه مشغول است             |
| ۱۰        | NLevt_ICV | نگارش شناسه ناسازگار است    |
| ۱۱        | NLevt_ACD | دسترسی غیر مجاز است         |
| ۱۲        | NLevt_IVP | VID یا رمز عبور اشتباه است  |
| ۱۳        | NLevt_DIS | شناسه غیر فعال شده است      |
| ۱۴        | NLevt_CAT | شناسه غیر فعال فجیع شده است |
| ۱۵        | NLevt_SUS | شناسه در حالت تعلیق است     |
| ۱۶        | NLevt_OTE | خطای ناشناخته               |



|   |                 |    |
|---|-----------------|----|
| سرور شناسه مشغول است                            | NLevt_NTBSY     | ۲۰ |
| سرور اتصال را قبول کرد                          | NLevt_NTAPT     | ۲۱ |
| سرور اتصال را رد کرد                            | NLevt_NTRJT     | ۲۲ |
| سرور شناسه پیدا نشد                             | NLevt_NTNAV     | ۲۳ |
| اتصال به سرور شناسه قطع است                     | NLevt_NTDC      | ۲۴ |
| اتصال به سرور وصل است                           | NLevt_NTCD      | ۲۵ |
| رشته وارده بسیار بزرگ است                       | NLevt_SBig      | ۲۶ |
| خطای سرویس وب (http)                            | NLevt_WCHttpEr  | ۳۲ |
| خطا در تفسیر دستور العمل                        | NLevt_WCCompEr  | ۳۳ |
| تنظیمات شبکه شناسه ست نشده است                  | NLevt_NETNS     | ۵۰ |
| کلید رمزنگاری AES ست نشده است                   | NLevt_KEYNS     | ۵۱ |
| رمز عبور کاربر (سطح ۳) تنظیم نشده است           | NLevt_UserPWDNS | ۵۲ |
| تنظیمات وب ست نشده است                          | NLevt_WebCNFNS  | ۵۳ |
| داده Unicode پذیرفته نمی شود                    | NLevt_UnicodeNS | ۵۴ |
| کلید رمزنگاری کاربر (کلید دوم) تنظیم نیست       | NLevt_UserKEYNS | ۵۵ |
| کاهنده خودکار (زمانسنج ۵،۵۹ ثانیه ای) فعال نیست | NLevt_TimerNS   | ۵۶ |

#### ۴- کاربردهای شناسه نوین

شناسه نوین کاربردهای مختلفی دارد که در ادامه به چندین مورد اشاره خواهیم کرد.

##### ۴-۱- قفل سخت افزار

ابتدایی ترین امکانی که در شناسه نوین تعبیه شد، استفاده از آن برای محافظت نرم افزار در برابر کپی های غیر مجاز است. محافظت نه به معنای جلوگیری از کپی کردن اطلاعات است بلکه تضمین عدم اجرا در عدم حضور شناسه مربوطه می باشد.

بدین منظور می توانید اطلاعات و پارامترهایی را که توسط نرم افزار برنامه ریز روی شناسه خام تنظیم نموده اید توسط متدهای موجود در کتابخانه رابط مورد بررسی قرار دهید.



برای بالا بردن ضریب امنیتی نرم افزارهای نوشته شده می‌توانید موارد زیر را رعایت کنید. حتما از رمز عبور استفاده کنید. رمز عبور را در نرم افزار برنامه ریز باید تنظیم نمائید و حتی المقدور رمز عبور را ۱۶ کاراکتر وارد نمائید.

کلید مربوط به کد گذاری AES را حتما تنظیم نمائید و در سورس نرم افزارهایتان حتما از روتین های Encrypt و Decrypt استفاده نمائید. ساده ترین روش برای این منظور این است که عددی را با این الگوریتم Encrypt نمائید. سپس عدد بدست آمده را در برنامه با روتین Decrypt رمز گشایی نمائید اگر عدد برگشتی با عدد اول برابر بود یعنی شناسه موجود است، همچنین می‌توانید به تعداد نامحدود عددهای مختلف را به شناسه ارسال و کد گذاری شده آن را بدست آورید (نرم افزار برنامه ریز این کار را برای شما انجام می‌دهد) به این صورت که شما تعداد عدد مورد نظر را وارد و سپس به تعداد مشخص شده عدد ایجاد و به شناسه ارسال می‌شود این عدد و عدد کدگذاری شده آن در یک فایل متنی ذخیره می‌شوند و حال شما می‌توانید این اعداد را در بین اطلاعات خود مخفی نمائید. به طور مثال در بانک های خود فیلدهائی اضافه و این اعداد را آنجا قرار دهید و یا می‌توانید کلیه اعداد را در یک فایل باینری با فورمتی که خودتان انتخاب می‌کنید قرار دهید و در نرم افزار به طور تصادفی عددها را خوانده به شناسه ارسال و جواب آن را با جواب موجود در بانک خود مقایسه کنید، در صورت برابر بودن مطمئن باشید که شناسه موجود می‌باشد.

علاوه بر این کلیدی را که داخل شناسه ست کرده اید را می‌توانید به صورت گسسته لابلای اطلاعات خود پنهان کرده و در هنگام تست شناسه اطلاعات تصادفی را هم با شناسه و هم با کلید داخل برنامه به روش Rijndael ECB Mode (AES) رمزنگاری کرده و نتایج را مقایسه نمائید.

از دیگر روش ها برای بالا بردن ضریب امنیت، استفاده از نرم افزارهای فشرده سازی برای فشرده کردن فایل اجرایی برنامه می‌باشد. این برنامه ها اطلاعات اضافی در فایل اجرایی را حذف می‌کنند و مزیت آن اینست که اولاً حجم فایل کم می‌شود، ثانياً فایل اجرایی غیر قابل Trace می‌شود.

ضمناً برای جلوگیری از دستکاری فایل اجرایی خود طول فایل را در هنگام اجرا چک کنید. همچنین برای شناسایی کتابخانه تقلبی NovinLock.dll که یکی از راههای دور زدن سیستم می‌تواند باشد با استفاده از کلاس CapiCom در ویندوز، امضای دیجیتال فایل مورد نظر را چک کنید که حتماً متعلق به شرکت نوین افزار باشد.

#### ۴-۲- قفل سخت افزاری تحت شبکه

بدین منظور کفایت نرم افزار اشتراک گزارنده قفل در شبکه را روی سیستم خاصی فعال کرد و پس از اطلاع از IP یا اسم کامپیوتر سرور و پورتی که در نرم افزار رابط تنظیم می‌گردد با استفاده از کلاس clsNetClient از دیگر کامپیوتر های از طریق شبکه به این نرم افزار وصل شد. پس از وصل شدن دقیقاً مانند یک قفل محلی می‌توان با آن رفتار کرد. در مورد کلاس فوق در بخشهای قبلی توضیح دادیم. در ادامه بحث به توضیح در مورد

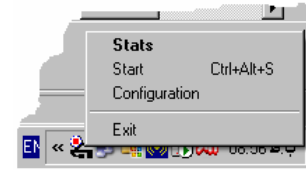


نرم افزار سرور شبکه قفل می‌پردازیم. این نرم افزار از دو قسمت تشکیل شده، یکی سرویس که در پشت صحنه ویندوز در حال اجرا است و دومی نماینده<sup>۱</sup> ای که وظیفه کنترل سرویس را به عهده دارد.

NovinLockNetAgent: نماینده این سرویس بعد از ورود به ویندوز به

صورت یک نشانه در کنار ساعت به صورت Tray نمایش داده می‌شود. این نماینده

علاوه بر کنترل سرویس از طریق منوی اصلی خود می‌تواند از طریق پارامترهای خط



فرمان نیز سرویس را کنترل کند:

/start: برای استارت سرویس مربوطه

/stop: برای ایست سرویس

/restart: برای راه اندازی مجدد سرویس

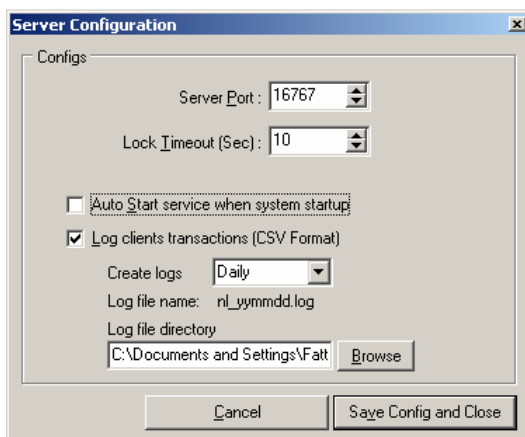
NovinLockNetServerSVC: این نرم افزار می‌زبان کاربران شبکه برای استفاده از شناسه شبکه می‌باشد:

این نرم افزار به صورت یک سرویس در ویندوز اضافه شده و استفاده از آن لزوماً نیازی به ورود (Login) به ویندوز ندارد.

برای تغییر تنظیمات این سرویس روی این نشان نماینده راست کلیک کنید تا منوی مربوطه نشان داده شود و مورد Configuration را انتخاب نمایید.

در فرم مربوطه ۶ مورد را می‌توان تنظیم کرد، این موارد در registry ویندوز ثبت شده و در دفعات بعدی

اجرا نیازی به تنظیم مجدد آنها نیست:



۱. Server port: پورت سرویس دهی به کاربران که

می‌تواند عددی بین ۱ تا ۶۵۵۳۵ باشد. ولی توصیه

می‌شود جهت عدم اختلال در کار سایر نرم افزار از

پورتهای استاندارد مانند ۲۱، ۸۰، ۱۳۵، ۱۳۹،

۴۴۵، ۱۴۳۳، ۳۳۸۹ و ... استفاده نشود. همچنین در

صورت استفاده از نرم افزارهای محافظ مانند Firewall ها استفاده از آن پورت را مجاز نمایید.

۲. Lock Timeout: در صورتی که هر یک از Client ها پیام اعلان موجودیت به سرور ارسال نکنند

بعد از این مدت سرور آنها را disconnect می‌کند.

۳. Auto Start ... باعث می‌شود که در هنگام بارگزاری ویندوز این سرویس به طور خودکار فعال

شده و آماده بهره برداری توسط کاربران شبکه شود.

۴. Log clients ... در صورت فعالسازی این مورد، نرم افزار از فعالیتهای کاربران روی شناسه گزارش

متنی تهیه می‌کند.

<sup>1</sup> Agent



۵. Create logs: تعیین می‌کند که نوع گزارشگیری چگونه باشد. به طور مثال در صورت انتخاب Hourly در هر ساعت یک فایل ایجاد می‌شود و گزارشها در داخل آن ریخته می‌شود.
۶. Log file directory: تعیین می‌کند که فایل‌های گزارشگیری در کجای سیستم ذخیره شود. ساختار این گزارشها به صورت زیر است:

تاریخ و ساعت شروع گزارشگری → #Date: 2006/05/14 21:20:37

#Fields: time c-ip c-port cs-command sc-result sc-bytes cs-bytes

فیلدهای اطلاعاتی به ترتیب از چپ عبارتند از:

زمان انجام / آدرس IP کاربر / پورت کامپیوتر کاربر / متدی که اجرا شده / کد نتیجه / مقدار بایت  
ارسالی از سرور / مقدار بایت دریافتی از کلاینت

### گزارش آماری از کاربران شبکه متصل شده:

با دوبار کلیک روی نشانه نرم افزار یا انتخاب Stats از منوی نرم افزار پنجره زیر باز می‌شود که نشان دهنده اطلاعاتی در مورد وضعیت نرم افزار، شناسه شبکه و کاربرانی که از آن استفاده می‌کنند را در اختیار ما قرار می‌دهد.

| Code | Client IP | Client Port | Connection Duration | Idle Time | Transactions |
|------|-----------|-------------|---------------------|-----------|--------------|
| 0    |           |             |                     |           |              |

Server Statistic window details:  
 Startup at: 2007/01/13 - 12:55:22 E.U  
 Capacity: --  
 Client Count: 0  
 Lock Serial: --  
 Lock Version: --  
 Memory Size: --

Buttons: Configuration, Disconnect Selected, Exit Stat

نکته ای که قابل توجه است که متد Start در clsNetClient زمانی درست انجام می‌شود که علاوه بر منطبق بودن Host و Port تعیین شده در آن با تنظیمات نرم افزار سرور باید شناسه ای که روی سرور قرار دارد برای شبکه ست شده باشد و همچنین ظرفیت تعداد کاربران آن نیز پر نشده باشد. در غیر این صورت clsNetClient خطای Connection rejected برمی‌گرداند.

برای درک بهتر موضوع به مثال این کلاس واقع در شاخه نمونه کدهایی که در شاخه Net Client هر زبان برنامه نویسی روی CD همراه شناسه بهره ببرید.



## ۴-۳- شناسه دیجیتال تحت وب

جهت استفاده از شناسه نوین برای شناسائی افراد و اعضای یک سایت اینترنتی به دو روش می‌توان از شناسه نوین استفاده کرد:

۱. ارتباط با شناسه از طریق صفحه کنسول
۲. شناسایی و تایید شناسه با استفاده از رمزنگاری AES به روش OTP<sup>۲</sup>

### ۴-۳-۱- ارتباط با صفحه کنسول

#### الف- پیش نیازها

شرایط لازم برای استفاده از این روش:

- نصب ActiveX مربوط به شناسه نوین (NovinLock.dll (NLLIB.clsNovinLock)
  - دسترسی به برنامه نویسی javascript یا VBscript در مرورگر وب کاربر
  - امکان دسترسی به این کلاس در برنامه نویسی وب (شرایط امنیتی مرورگر نباید محدود شده باشد)
- عناصر دخیل در فرآیند انجام کار
- وب سرور (هیچ محدودیتی در نوع یا سیستم عامل یا زبان برنامه نویسی وجود ندارد)
  - صفحه وب کنسول (تبادل اطلاعات بین سرور و کاربر را انجام می‌دهد)
  - مرورگر وب (که ActiveX را فرمان می‌دهد)
  - clsNovinLock (واسط بین سیستم و شناسه است)

#### ب- مراحل اجرای فرآیند

مراحل کلی برای اجرای یک سری از متدهای clsNovinLock از طریق وب به روال زیر است:

۱. ایجاد متن دستورات سمت سرور و ذخیره آن در یک واسط (مثل DB یا Application)
۲. تخصیص یک کد رفرنس یکتا به متن دستورات عمل
۳. اعلان رفرنس آن به شناسه از طریق script موجود در صفحه وب که در حال ارسال به کاربر است
۴. مراجعه clsNovinLock به وب سرور و دریافت لیست دستورات
۵. اعلان clsNovinLock به وب سرور جهت دریافت صحیح دستورات برای آزاد کردن منابعی که آن دستورات عمل آنها را اشغال کرده بود
۶. تفسیر و اجرای دستورات عمل توسط clsNovinLock که توسط یک موتور vbscript انجام می‌شود
۷. ثبت نتایج بدست آمده در سرور توسط clsNovinLock
۸. استخراج مقادیر و نتایج لازم از نتیجه ارسالی در سرور

<sup>2</sup> One Time Password



## ج- ساختارها و قوانین

### ۱. ساختار دستورالعمل:

کاراکترهای کنترلی در این دستورالعمل سه عدد می‌باشد :

% برای جداسازی قسمت های مختلف دستورالعمل

# در ابتدای هر قسمت نشان از وجود یک مقدار بازگشتی در آن قسمت است ! در ابتدای هر قسمت باعث می‌شود آن قسمت فقط اجرا گردد.

اولین قسمت دستورالعمل نشان دهنده ReferenceID مربوط به ثبت نتایج است هنگامی که از متدهای خود ActiveX می‌خواهیم استفاده کنیم از کلاس NL باید استفاده گردد. به طور مثال برای دریافت سریال شناسه باید از NL.GetSerial بهره برد .

قسمتهایی که با # یا ! شروع می‌شوند، باید از ساختار VBScript پیروی کنند.

### ۲. ساختار صفحه کنسول:

این صفحه وظیفه دارد بر اساس رفرنس ارسالی از clsNovinLock و سوئیچ مربوطه عمل متناسبی را انجام دهد . سه وظیفه اصلی این صفحه عبارت است از :

- Get Command: برای نشان دادن متن دستورالعمل به شناسه
- Kill Command: برای آزادسازی منابع مربوط به آن دستورالعمل
- Set Result: جهت دریافت نتایج دستورالعمل از شناسه و ثبت آن با رفرنس ارسالی

برای تشخیص سه پارامتر فوق باید از کلید Action استفاده کرد (مانند Request("Action")) سه کلمه دلخواه برای این سه پارامتر تعیین کرده و آنها را به هنگام برنامه ریزی شناسه ها در تب Net and Web options باید وارد نمایید.

برای دستیابی به رفرنس ارسالی نیز با کلید referID می‌توان کار کرد (Request("referID")) نمونه کد کنسول به زبان ASP:

```

<%
RefID = Request("ReferID")

Select Case Request("Action")
    case "GetCMD":
        Response.Write(Application(RefID))

    case "KillCMD":
        Application.Contents.Remove(RefID)
        Response.write "Destructing"

    case "Resolver":
        Application(RefID) = Request("Results")

    case else
        Response.Write "Error " + Request.QueryString
End Select
%>

```

با توجه به نمونه کنسول بالا کلمات کلیدی برای ۳ عمل اصلی عبارتند از :



|                      |                   |
|----------------------|-------------------|
| <b>Get Command:</b>  | <b>"GetCMD"</b>   |
| <b>Kill Command:</b> | <b>"KillCMD"</b>  |
| <b>Set Result:</b>   | <b>"Resolver"</b> |

#### د- برنامه ریزی شناسه برای وب

در تب Net and Web options در نرم افزار برنامه ریز قسمت مربوط به وب را با توجه به کلمات کلیدی تعریف شده در صفحه کنسول و محل قرار گرفتن آن در سایت را در فیلدهای مربوطه وارد کنید. توجه داشته باشید در این قسمت مجموع موارد وارد شده حداکثر می تواند ۷۶ کاراکتر باشد. پس از فعال سازی شناسه وب ۸۰ بایت از فضای داده شناسه کم خواهد شد.

مثال مربوط به این روش در شاخه Samples/ASP/Nlock وجود دارد.

#### ۴-۳-۲- روش OTP

در این روش نیاز هست که از امکانات رمزنگاری دوم شناسه استفاده شود.

مراحل اجرای این روش به صورت زیر است:

۱. تنظیم رمز عبور سوم (کاربر) و کلید رمزنگاری دوم (شخصی) با استفاده از متدهای SetUserPWD

و SetUserKey

۲. ایجاد یک بانک اطلاعاتی شامل فیلدهایی برای شماره سریال شناسه، اطلاعات آشکار، اطلاعات

رمزنگاری شده و یک flag برای عدم استفاده مجدد از رکورد

۳. پر کردن بانک با استفاده از رمزنگاری نوع دوم با شروع از یک رشته شانسی و به صورت تکراری

پشت سر هم به تعداد لازم (مثلا ۵۰ عدد)

۴. در هر مرحله از چک کردن شناسه مثلا در هنگام Login به وبسایت یکی از اطلاعات آشکار که تا به

حال استفاده نشده، براساس شماره سریال شناسه که از طرف کلاینت به سرور ارسال شده انتخاب و

به مرورگر وب کلاینت ارسال می گردد.

۵. دریافت رمز عبور سوم از کاربر و انجام عمل رمزنگاری و ارسال آن به سرور

۶. در صورتی که جواب رمزنگاری با جوابی که قبلا در بانک ذخیره شده بود برابر بود. آن رکورد برای

عدم استفاده در آینده علامت می خورد و به کاربر اجازه ورود به وبسایت داده می شود

۷. در صورتی که رمزنگاریها در بانک تمام شد، یعنی همه موارد مورد استفاده قرار گرفت می توان با

استفاده از روش کنسولی یک سری رمزنگاریهای دیگر را جایگزین کرد

توجه داشته باشید که این روش بیشتر در ورود به وبسایت یا در هنگام انجام یک عمل حساس استفاده

می شود و در حرکت در لابلای صفحات استفاده نمی شود.

جهت درک بهتر موضوع به مثالی که در شاخه Samples/ASP/OTP هست توجه نمایید. علاوه بر پیاده

سازی صفحات وب، نرم افزار جانبی برای تنظیمات اولیه به زبان دلفی نیز در این شاخه وجود دارد.



#### ۴-۴- امضای دیجیتال

امضا در واقع تایید صحت اطلاعات و قبول مسئولیت اطلاعات است. به طور مثال زمانیکه شما برگه ای را امضا می‌کنید در واقع علاوه بر تایید اطلاعات از طرف خود، مسئولیت اطلاعات مندرج در آن برگه را نیز به عهده می‌کشید.

امضا در دنیا مجازی کامپیوتر و اینترنت یک امکان دیجیتالی است که منجر می‌شود به:

۱. از انکارهای احتمالی کاربر در قبال تراکنش انجام گرفته جلوگیری شود

۲. از دستکاریهای مستقیم که با دسترسی های غیرمجاز انجام می‌گیرد جلوگیری شود

بدین منظور از رمزنگاری اطلاعات و نگهداری آن در پیوست اطلاعات اصلی استفاده می‌شود.

از آنجایی که شناسه نوین توانایی رمزنگاری اطلاعات با دو کلید ۱۲۸ بیتی را دارد می‌توان از آن برای این منظور استفاده کرد. یعنی کلید اول که معمولاً به صورت همگانی تنظیم می‌شود برای اطمینان از مورد ۲ و کلید شخصی یعنی کلید دوم برای مورد ۱.

علاوه بر این می‌توان از الگوریتم RSA (رمزنگاری نامتقارن) استفاده کرد و از شناسه نوین به عنوان وسیله ای برای نگهداری کلید خصوصی (Private Key) استفاده کرد. زیرا که یکی از مشکلات RSA نداشتن مهارت لازم کاربران برای نگهداری از کلید خصوصی خود می‌باشد.

#### ۴-۵- مجوز ورود به ویندوز



Please Insert Your Novin Token  
Or Press Ctrl + Alt + Del For Normal Login



یکی از کاربردهای عمومی شناسه نوین استفاده از آن برای ورود به سیستم عامل ویندوز است. برای اطلاعات بیشتر به مستندات ویژه این کاربرد در شاخه Windows Login در CD همراه شناسه مراجعه نمایید.

**شناسه نوین گامی در جهت تحقق اهداف دولت الکترونیک**