MXLOGON2SC MANUAL

2-Factor, 2-Step Windows Sign-in With USB Smart Card Token

https://ribig.co.jp/mxlogon2sc

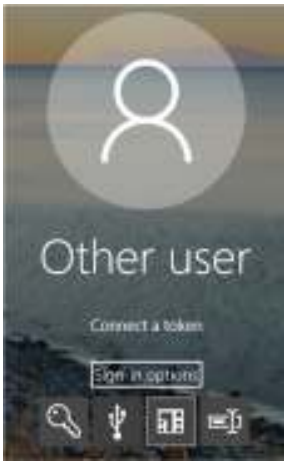
DECEMBER, 2018

RIBIG INC.
mxlogon2@ribig.co.jp

Contents

1. About MxLogon2sc

It is a 2-factor and 2-step Windows sign-in solution. To log in via MxLogon2sc, you must have a valid a smart card USB token.



Once you plug in your token, you prove that you are the rightful owner of the token by entering the current PIN of the token. Successive entries of false PINs will lock the token.

When the 2-factor authentication with the token is successful, you will, then, be asked to enter username/password for the second authentication.

Your token may be configured to specify which user can log in in the second step. Trying to log in under different users with the token results in an error.

For those who can trade off security for convenience, MxLogon2sc has the feature to auto-fill PIN, username and password, thus requiring no manual entries. When PIN auto-fill is enabled, the PIN entry screen will not be shown. With username auto-fill, the username field in the second step will have the specified username entered in advance. Username/password auto-fill will automate the second step authentication.

With PIN and username auto-fill, you will only see the username/password authentication screen with the username already completed. With PIN and username/password auto-fill, having a token connected will log you in automatically.

After you logged in with a token, unplugging it will lock the login session.

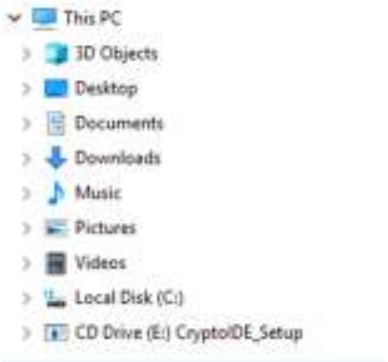
Depending on how you use MxLogon2sc, it can be a highly secure Windows login solution or a modestly secure and undemanding way to sign in conveniently to Windows.

2. USB Smart Card Token & Driver

Before installing MxLogon2sc, connect a USB token that comes with MxLogon2sc to the computer to install it on. Windows will detect the toke and automatically load an appropriate driver. To see if the token is ready for use, run “changepin.exe” in x86 folder of the distribution file.

The list box on the left should have the token’s serial number if it is available for use with this software.

If the token is not detected, run the setup program on CD-ROM area of the token. It has a built-in CD-ROM area. When it is connected, Windows pops up a window about the newly connected drive. You can find the drive in Explorer (the drive letter can be different)



MxLogon2sc does not rely on the software contained in the CD-ROM; it accesses the token solely through PKCS#11 module bundled with this software. But installing the setup program may help in case the token is not detected.

MxLogon2sc token is a general-purpose smart card token. You can use it for use with other programs that work with smart cards, such as BitLocker, VPN and Web access. You will need

the utility programs on CD-ROM to get the token available for these applications. But be sure that you do not delete the data or certificate that are preset or MxLogon2sc creates. Doing so will affect MxLogon2sc.

3. Remote Desktop & RDP Redirect

You can sign in to a remote computer with MxLogon2sc installed, using a token connect to the local computer. Microsoft remote desktop client can redirect the locally connected token to the remote computer. This has the effect that the remote computer detects the remote token as if it were connected locally. Since MxLogon2sc on the remote side can access the remote token like a locally connected token, you can use the local token to log in via remote MxLogon2sc.

If you plan to log in to a remote computer with MxLogon2sc token, check to see if RDP redirect is enabled and a local token is redirected to a remote computer.

How to check:

1. Copy “changepin.exe” and “cryptoide_pkcs11.dll” in x86 folder of the distribution file to the remote computer
2. Connect to the remote computer using Microsoft remote desktop client.
3. Run “changepin.exe” on the remote side.

When the remote side detects the local token, the list box on the left should have its serial number. Try plugging off and on the token and see if the list box responds to the event correctly.

4. Installing MxLogon2sc

If you have downloaded the distribution ZIP file, unblock it first. In Explorer, right-click the zip file and select “Property”. If the bottom part of [General] tab shows “unblock” check box, check the box and press [OK].

Run “auto-setup.exe” in the root folder of the distribution file.

Click on [Install] button. The installation will normally complete in a few seconds. When finished, restart the computer.

5. Signing in via MxLogon2sc

In the sign-in screen, click on “Sign-in option” and select MxLogon2sc icon.

When a token is connected, the screen shows PIN field.

When multiple tokens are plugged in, you can select a token in the combo box.

The default PIN is **"12345678".**

Once the token is authenticated, the screen changes to one for username/password authentication.

You can enter any valid combination of username and password to sign in.

Pin Lock

By default, when you enter 10 false Pin successively, the token will be locked. Unless SO user unlocks it, PIN authentication will always fail. SO user can unlock a locked token using “cert.exe” utility in MxLock2sc package.

6. Screen Lock

After you have logged in with a token, unplugging the login token, while being logged in, will lock the login session. If multiple tokens are connected at the login screen and you select one of them, the token used for sign-in will lock the screen; plugging off other tokens will have no effect.

7. Signing in to a remote computer and Locking

You can sign in to a remote computer using Microsoft remote desktop client, using the locally connected token. This assumes that the local token is redirected to the remote computer by RDP Redirect. Unless the token is redirected, the remote login by the local key will not work.

When NLA (network level authentication) is enabled, you enter the remote computer's credential locally. MxLogon2sc on the remote side will show the PIN entry and once the token is verified, MxLogon2sc will sign you in automatically, using the credential received from the connecting side.

MxLogon2sc may behave differently when PIN/username/password auto-fill is set. It gives auto-fill higher priority than the credential received from the client side.

8. Token Configuration

MxLogon2sc smart card token and software are configured differently for each user. The software for a user will not detect any tokens for other users; it works only with the matching tokens. But this is not enough for safe use of the tokens.

Mandatory token configuration:

- All MxLogon2sc tokens for any users will have the same default user PIN "1234555678" as well as the same SO PIN "admin1234". Unless you change them, anyone who gets hold of a token can have it authenticated in the first step.

Optional token configuration:

- You can save a certificate signed by your own CA to each token and have MxLogon2sc verify the certificate against CA. MxLogon2sc programs property set up will not detect a token without a valid certificate.

Use the token configuration program, "cert.exe", in the distribution file. This program is not installed to any computers. Anyone who has this program can configure your tokens; take an extra care to guard this program from unauthorized copy and use.

You cannot run "cert.exe" from a read-only folder.

1. Copy “config” folder in the distribution file to a folder in a fixed disk.
2. Grant read and write access only to the token administrators. Revoke all rights to this folder from any other users.
3. Destroy the original “cert.exe” later when you no longer need it.

[Config] folder contains the following files and folders:

Cert.exe

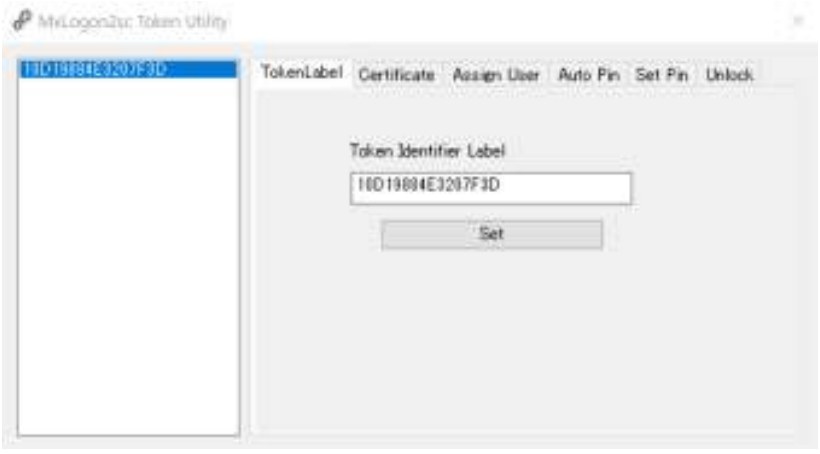
cryptoide_pkcs11.dll

[openssl] folder

+ openssl.cnf
+ openssl.exe
+ libcryptoMD.dll
+ libsslMD.dll

8.1 Token Identification String - optional

You can assign a human readable string to a token to help you recognize it from the others. The default string is the token serial number. If you connect a token only one at a time, then, you may not need a way to identify a token.



Select a token to set a string to on the left list box, set an identification string and click on [Set] button.

8.2 Certificate - optional

You may save a certificate to a token to bind it strongly to your organization. A token without a valid certificate will never have any chance to get authenticated by MxLogon2sc for the organization. This certificate is used only for the token authentication in the first step, not for the user authentication in the second.

In [Certificate] tab, “cert.exe” creates CA and issues certificates using openssl.exe command. Users familiar with creating CA and issuing certificates signed by the create CA can do so without using this program. A token utility on CD-ROM can import .pfx file to a token.

In this tab, you create a CA first.

Fill in the fields for CA certificate and press [Create CA] button.

A window pos up for the password to protect CA’s private key. Enter one and press [OK].

CA will be created and, if successfully, “CA created” window pops up.

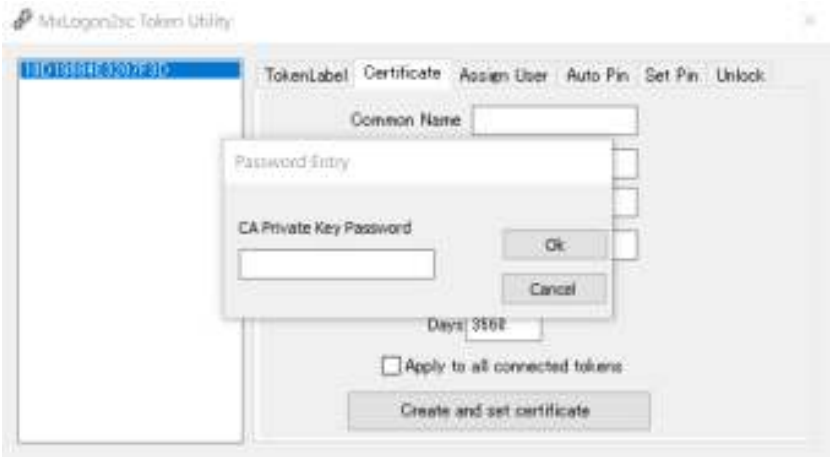
Once CA is created, the tab is for issuing certificates signed by the created CA. The field data come from the ones you entered for creating CA. You may change them in any way you need.

[Create and set certificate] will create a key pair, sign the public key by the CA and save the certificate and the private key to the selected token.

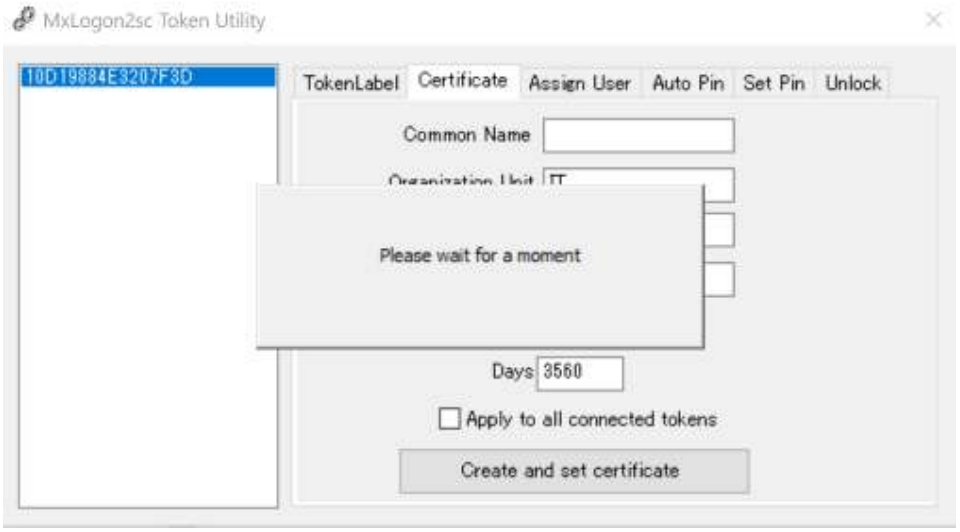
A valid certificate must have “Common Name” set to the token’s serial number. Leave “Common Name” empty and the serial number is automatically set.

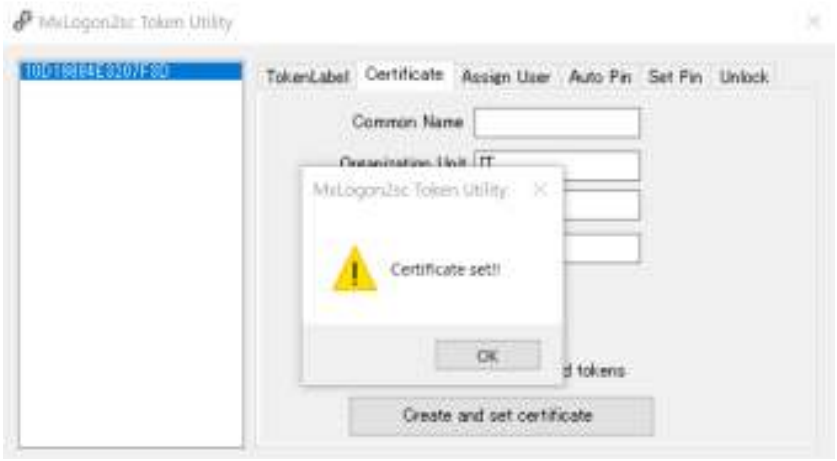
Th certificate validity duration “Days” is important; when the token certificate expires, MxLogon2sc will not detect the token.

Select a token to save the issued certificate to and press [Create and set certificate] button. You will be prompted to enter the password for CA private key.



Pressing [OK] will start creating a certificate and saving it to the selected token.





Creating and setting certificates to multiple tokens

When you connect multiple tokens and enable “Apply to all connected tokens”, clicking on [Create and set certificate] button will repeat the process of creating a certificate and saving it to each connected token.

Creating multiple CAs

Cert.exe runs “openssl.exe” in [openssl] folder inside the folder where “cert.exe” exist. You copied [config] folder to a folder on a fixed disk. When you copy [config] folder to another folder and run “cert.exe” in the new folder, it will find no CA file. You can create another CA and issue certificates by the new CA.

[Config]

- Cert.exe
- cryptoide_pkcs11.dll
- [openssl] folder
  - + openssl.cnf
  - + openssl.exe
  - + libcryptoMD.dll
  - + libsslMD.dll

Enabling the certificate verification

Storing a certificate to a token does not automatically enable the certificate verification. MxLogon2sc needs to know which CA certificate to use for the validation. You can find the CA certificate file in [openssl] folder as “cacert.pem”. Copy it to MxLogon2sc installation folder.

The installation folder is probably “C:\Program Files\RiBiG\MxLogon2sc”(to be more correct, it is %ProgramFiles%\RiBiG\MxLogon2sc). When MxLogon2sc find “cacert.pem” in the installation folder, it will try to verify the designated certificate in the token. It is treated as if it were not detected when

- no certificate is found, the token is treated as if it were not detected.
- the certificate is not signed by the CA.
- the certificate is expired.
- Common Name of the certificate is different from the token serial number.

Or during PIN verification, you get PIN error when no matching private key is found.

Using the token certificate for accessing protected Web site - optional

Web directory may be configured to allow access only to those users who can present valid client certificates issued by a designed CA.

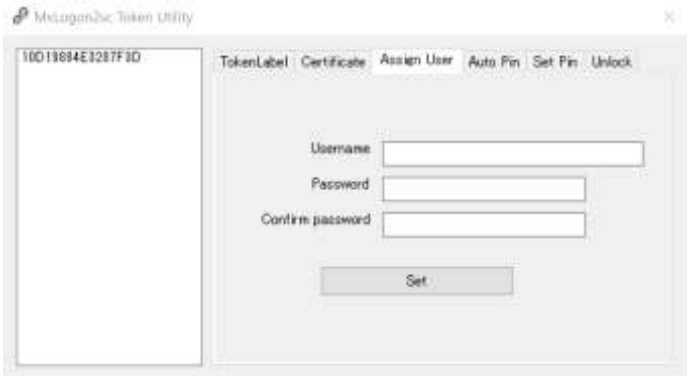
Apache Example

1. Install the utility program on the token’s CD-ROM
2. Copy “cacert.pem” to a web server and set its path to “SSLCACertificateFile”
3. Protect a web directory requiring valid certificates

```
<Location /protected>
    SSLOptions +StdEnvVars
    SSLVerifyClient require
</Location>
```

8.3 Assign User Tab - optional

This tab enables the credential auto-fill.



- Username auto-fill – set username text to auto-fill username field in the login screen
- Username/password auto-fill – set both username and password
- Setting password alone is not permissible

Click on [Set] button

To specify a local user, use "." (dot) for the local domain name. MxLogon2sc on each computer translates it to its computer name.

Deleting auto-fill

Leave username field empty and press [Set] button.

Auto password change

When you enable username/password auto-fill, the token user does not have to enter username nor password. The user does not have to know them or does not know them from the beginning. We assume that, when username/password auto-fill is enabled, the user cannot change passwords because the user does not know the old one.

When a token with username/password auto-fill enabled is used for sign-in, MxLogon2sc will change the passwords automatically, generating a new one.

While logged in with a token, pressing CTRL+ALT+DEL and selecting “Change a password” will change Windows passwords and update the auto-fill password in the token, without asking to enter the old password or a new one. The same is true when you try to sign in through MxLogon2sc but the password is expired. Windows asks you to change passwords and MxLogon2sc will change the passwords automatically. Since new passwords are generated, there is no way to know them.

Keep pressing SHIFT+CTRL will prevent a new password from being generated. The change password screen will have the old password auto-filled and you can enter the new password manually

.

8.4 Pin Auto-Fill - optional

Set the token’s PIN for auto-fill. Once you set PIN auto-fill, MxLogon2sc does not show the first step PIN entry screen.

To disable PIN auto-fill, leave Auto Pin field empty and press [Set Auto Pin] button.

8.5 Changing PIN - mandatory

Cert.exe assumes that user PIN and SO PIN of the token to be configured are set to their default value; otherwise, the program must ask user PIN or SO PIN of each token every time it attempts to configure it.

Default SO PIN	:	“admin123”
Default User PIN	:	“12345678”

Before configuring tokens. be sure to reset their SO and user PIN to the default values.

Once a token is configured, change SO and User PIN to operational one. Never leave the default PINs unchanged.

In [Set Pin] tab, you can change SO Pin / User Pin of all connected tokens.

SO Pin Change

Enter the current SO PIN. The default SO PIN (admin123) is always set but you need to set the correct SO PIN. Also set new SO PINs. Clicking on [Set SO Pin] will change SO PIN of the connected tokens.

User PIN Change

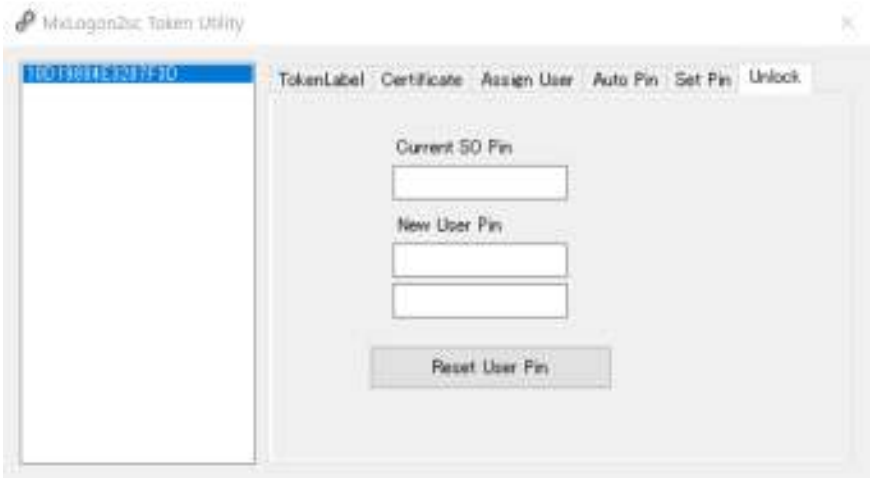
Enter the current user PIN. The default User PIN （" 12345678" ）is always set but you need to set the correct User PIN. After setting new User PINs, press [Set User Pin].

PIN Change Log

The PIN changes are logged to a file “keypin.log” in the folder where you can find “cert.exe”. If you lose SO PIN, there is no way to recover it. The manufacture hardware reset can restore the token back to the operational state, but it is costly. The log will help keep track of SO PIN of each token.

8.6 Unlock – when necessary

Use this tab to reset the locked token. This is a privileged operation and you need to provide SO Pin of the token to unlock.



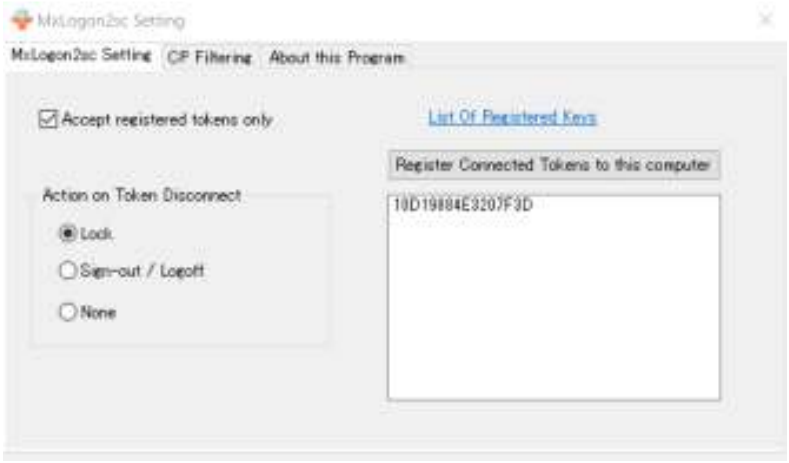
9. MxLogon2sc Configuration

Run “config” to configure MxLogon2sc. The program requires the administrative privileges for execution. You can find the program in [Start]-[MxLogon2sc]. The file is in MxLogon2sc installation folder. %ProgramFiles%\RiBiG\MxLogon2sc\config.exe

9.1 Setting

Action on Token Disconnect

By default, MxLogon2sc locks the screen when the login token is removed while logged in. You set it to sign you out or to do nothing by setting this option.



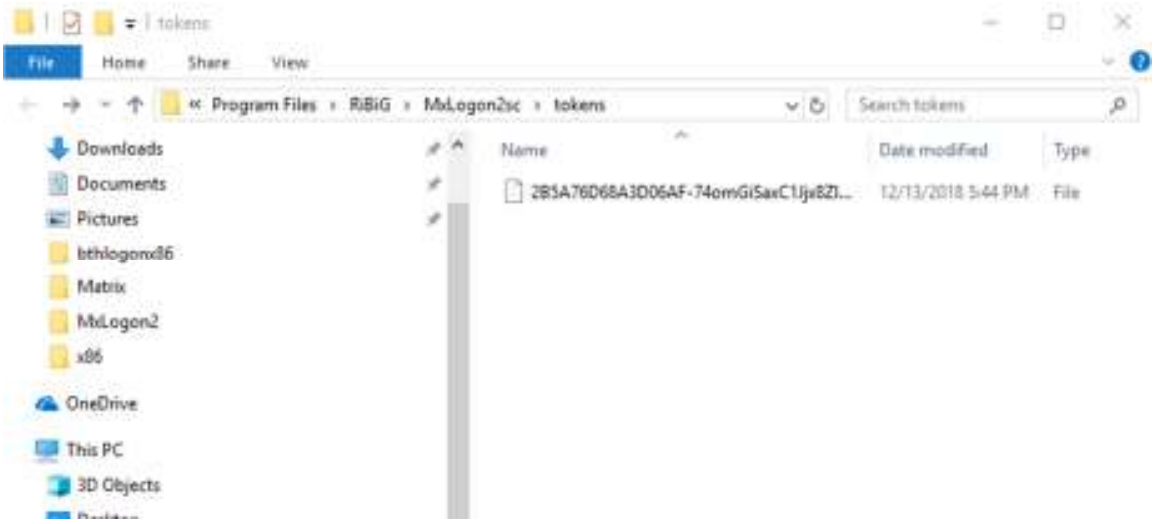
Accept registered tokens only

When enabled, MxLogon2sc will discover the valid tokens that are registered on this computer. It will also show the token registration button, link and list box.

- When no token is registered, MxLogon2sc ignores this option and accepts any valid keys.
- Tokens whose identification string ends with “-master” will not have to be registered. MxLogon2sc will consider such a token as “master”.

“List of Registered Keys” Link

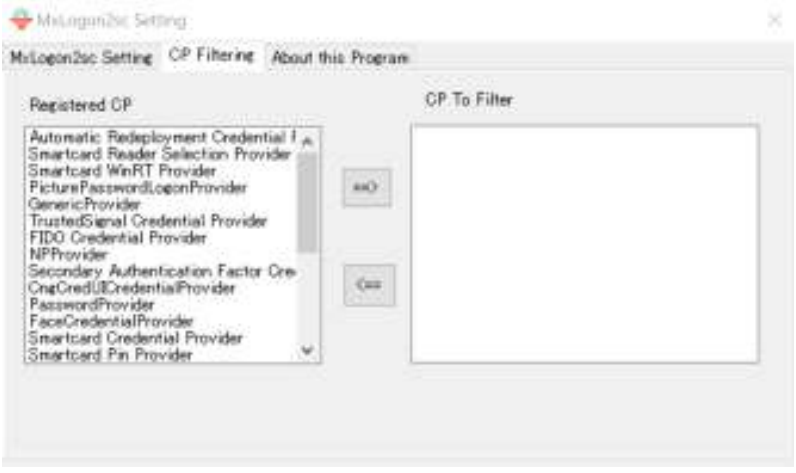
Tokens are registered in [tokens] folder of MxLogon2sc installation folder. This link opens the folder in Explorer. De-register a token by deleting a file corresponding to the token in the folder. The first part of the file name is the token identification string.



[Register connected tokens to this computer] Button

This will register the tokens shown in the list box on this computer and files for the tokens are created in [token] folder.

9.2 CP Filter Tab



When a credential provider is filtered, it will be hidden. You cannot select it where you can normally find it. After installing MxLogon2sc, you may want to require users to have a valid token to sign in to Windows. For that, you need to filter,

PasswordProvider

PicturePasswordLogonProvider

PINLogonProvider

WLIDCredentialProvider

Smartcard Credential Provider

On a computer joined to Azure AD, filtering

NGC Credential Provider

will hide PIN logon provider in the login screen.

To filter, select the one you want to filter in the left list box and press [=>] button to move it to the right list box.

Safe Mode

MxLogon2sc does not work in Safe Mode. If you need to protect Windows login in Safe Mode, we have a solution called SimplePCLock which prevents any login attempts when no valid USB memory stick is connected.

https://ribig.co.jp/simplepclock

10. Changing User Pin

A utility, “changepin” is available to help users to change their token PIN.

You can find it in [Start]-[MxLogon2sc] or

at ProgramFiles%¥RiBiG¥MxLogon2sc¥changePin.exe.

Once run, select the token whose user PIN to change, enter the current User PIN and new PINs and click on [Change Pin] button.

It will also update the token password when username/password auto-fill is enabled.

11. Uninstalling MxLogon2sc

Uninstall MxLogon2sc in [Setting] - [Apps & Feature] or in Control Panel.

12. OpenSSL License

OpenSSL License

/*

==========

=====

* Copyright (c) 1998-2017 The OpenSSL Project. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in

* the documentation and/or other materials provided with the

* distribution.

*

* 3. All advertising materials mentioning features or use of this

* software must display the following acknowledgment:

* "This product includes software developed by the OpenSSL Project

* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

*

* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to

* endorse or promote products derived from this software without

* prior written permission. For written permission, please contact

* openssl-core@openssl.org.

*

* 5. Products derived from this software may not be called "OpenSSL"

* nor may "OpenSSL" appear in their names without prior written

* permission of the OpenSSL Project.

```
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 *
================================================================
=====
 *
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com).  This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

 Original SSLeay License
 -----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
```

* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are adheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*   Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must include an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

```
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed.  i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/
```