USING DIGITAL CERTIFICATES FOR MAX AUTHENTICATION

How to obtain and use a digital certificate that allows you to access MAX services from scripts and batch jobs

If you want to use a program (such as a script or a scheduled task) to do things in MAX.gov, then you need a way for the script to identify itself to MAX. As an interactive user, you would normally use your PIV card or a username and password, but a non-interactive script doesn't have that ability. The solution is to use a digital certificate, which functions much like your PIV card. This document tells you how to do that. By the time you're done, you will know

- What a digital certificate is,
- The steps required to request and obtain a digital certificate for use with MAX,
- How to take care of the digital certificate once you have it and
- How to use the digital certificate in a script that accesses a MAX site.

If you're interested in a detailed discussion of security and privacy controls that pertain to the use of digital certificates, you can read NIST Special Publication 800-53.[1]

Scope

This document applies to anyone who needs a service ID and a digital certificate to represent it, including MAX staff and agency users.

Policy Reference

Please consult MAX Security Identification and Authentication Policy for a detailed policy statement. Note that as with other MAX documents referenced herein, you will have to have a MAX account to view this document.

Roles and Responsibilities

- Service ID User – that's you. You need a service ID and credentials for it in order to access a MAX system from a non-interactive script, and this document tells you how to do that.
- MAX Point of Contact (POC) – a person on the MAX team who is your advocate in the process, and who can work with MAX staff to complete your request.
- MAX Data Management Team – issues service accounts to the MAX POC.
- MAX Technology Services Team – issues digital certificates for service accounts to the MAX POC based on service account ID and a Certificate Signing Request (CSR) that you will generate.

What is a digital certificate?

In the course of working with systems and web sites, you will run across the term "digital certificate" a lot. Digital certificates are used for a number of different things – the most common use you will see is as an identification and encryption mechanism when you visit a secure (https) web site. Another common place where you'll encounter digital certificates is on your PIV card. Like an SSL certificate, the certificate on your PIV card contains an "identity" and a set of encryption keys. Unlike an SSL certificate, however, your PIV certificate is marked as being used "to identify a client", rather than "to identify a web site". Other than that, though, they're basically the same thing.

[1] Available as of this writing in draft form at https://csrc.nist.gov/csrc/media/publications/sp/800-53/rev-5/draft/documents/sp800-53r5-draft.pdf.

For our purposes, we can think of a digital certificate as a file that contains, among other things,

- The MAX ID that it represents,
- A unique identifier,
- An expiration date,
- A digital signature that ensures the integrity of the information,
- A reference to the authority that was used to issue the certificate,
- A reference to an authority that can be used to check the validity of the certificate and
- A public key that is used in cryptography[2], and an optional private key.

Most of the information contained in a certificate is public; that is, it's viewable by anyone. The exception is that a certificate used for identification must also have a private key. The private key, which is carefully guarded, is what allows the certificate to be used as an identity. Just FYI, your PIV card contains the private key used to establish your identity, and that private key is protected by your PIN (and some logic on the card itself that makes it difficult to guess your PIN). A MAX identify certificate works in just the same way, minus the PIN and associated card logic.

As we'll see later, you will use a MAX identify certificate by attaching it to a web request that causes MAX to treat your request as if it had been authenticated using a PIV card.

> **IMPORTANT**: A digital certificate with its private key are valid login credentials to MAX, and must be treated with care. Be sure to follow the practices below to obtain, protect and use your MAX digital certificate.

Getting a MAX Certificate

There are two main steps in getting a MAX certificate:

1. Request a service ID. Note that you can't use your own MAX ID in a certificate like this; it has to be a separate ID (a policy requirement).
2. Create a certificate request, and have it fulfilled by the MAX staff. There are different procedures for creating the certificate request, depending on whether you're using Windows or Linux.

If you're not a member of the MAX staff, then MAX POC will assist you with step 1, and will assist with the fulfillment part of step 2.

Getting a Service ID

Please ask your MAX POC to request a service ID from the MAX Data Management Team.

[2] See https://docs.oracle.com/cd/E19509-01/820-3503/ggbgc/index.html for a simple explanation, and https://en.wikipedia.org/wiki/Public-key_cryptography for a more extensive discussion of public key cryptography.

For MAX Staff: Please specify the following in your service ID request to the DMT:

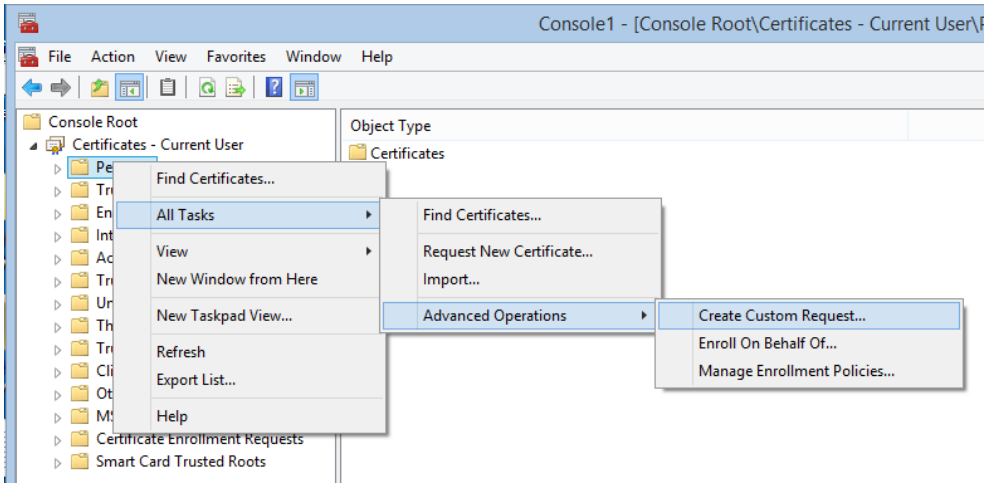
Service ID: (e.g., “S_MYSVC”)
Email: (e.g., “myservice@max.gov”)
First Name: (e.g., “My”)
Last Name: (e.g., “Service”)
Agency: (e.g., “TREASURY”)
Comment: (e.g., “This account will be used for …”)

Once you have established a service ID, the next step is to generate a Certificate Signing Request (CSR) and have the MAX staff fulfill that request for you.

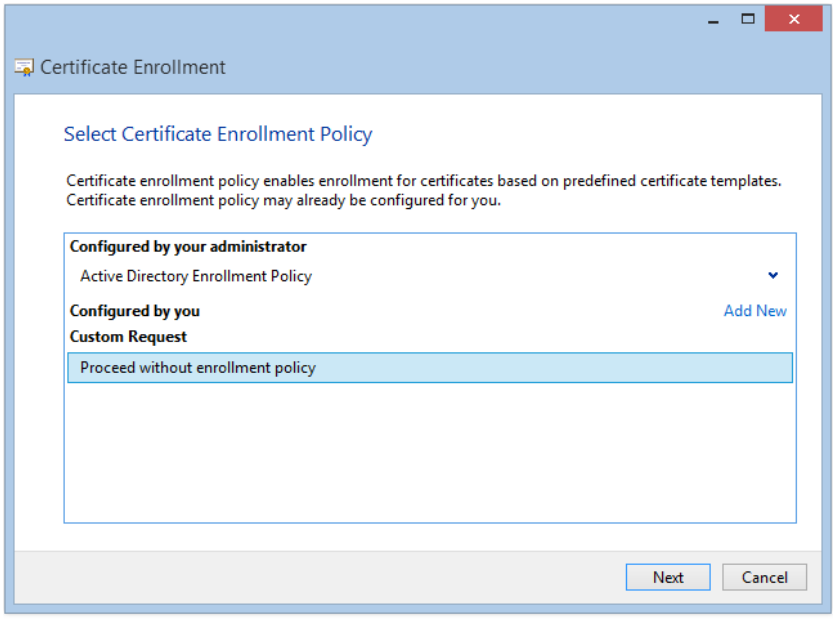
Getting a client certificate on Windows

Once you have established a service ID, you can request a certificate that represents that identity. Here are the steps:

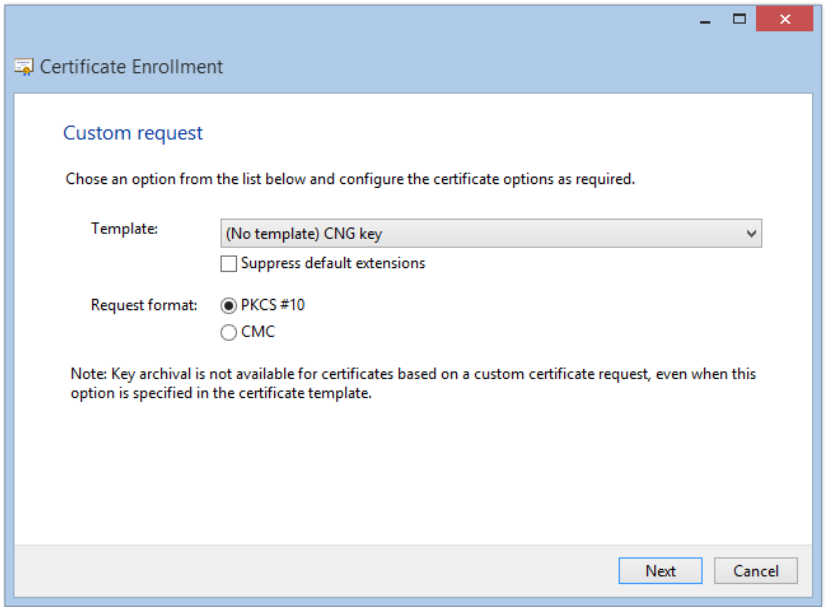
1. Create a certificate request on your system.
    a. Run “mmc” by hitting the Windows key, then typing “mmc”.
    b. In mmc, select File>Add/Remove Snap-in
    c. Select “Certificates”, and if you have the option, select “current user”.
    d. In mmc, select your personal store, then right-click, “All Tasks”, “Advanced”, “Create Custom Request”:



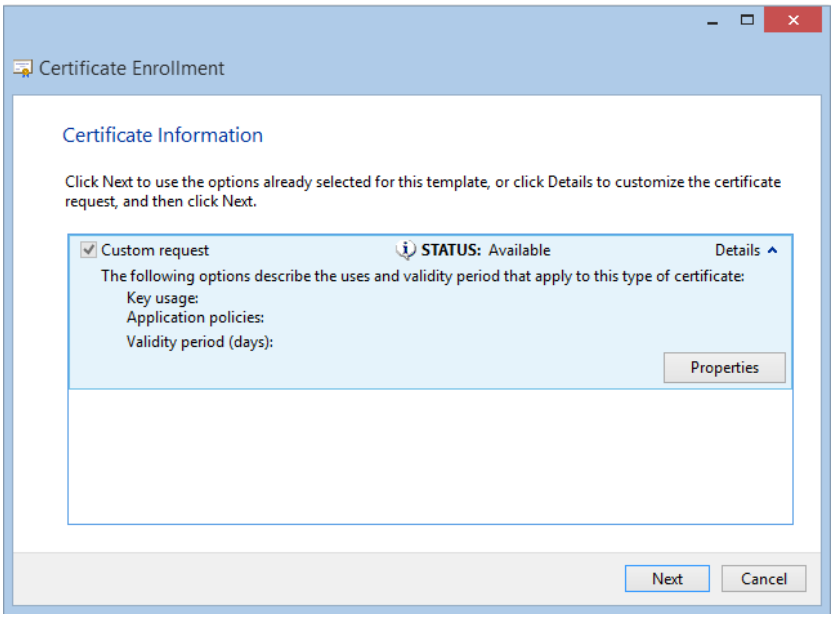
e. Click through the introductory screen(s), then pick “Proceed without enrollment policy”:



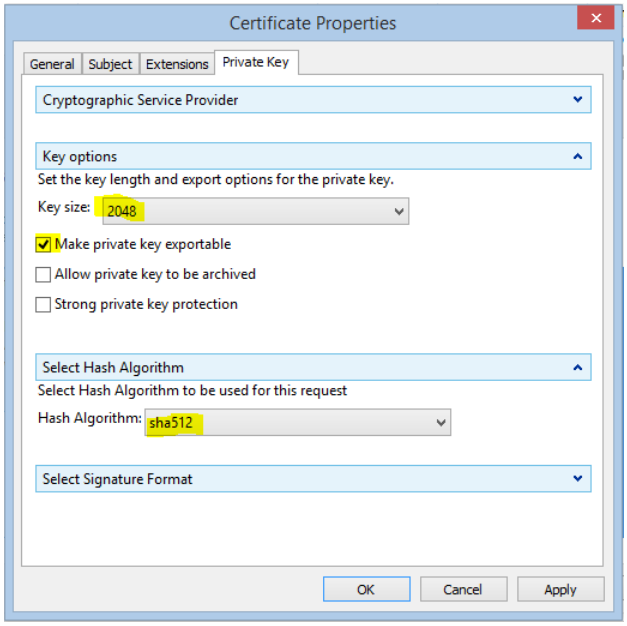
f. On the next screen, pick the options “CNG Key” and “PKCS#10”:



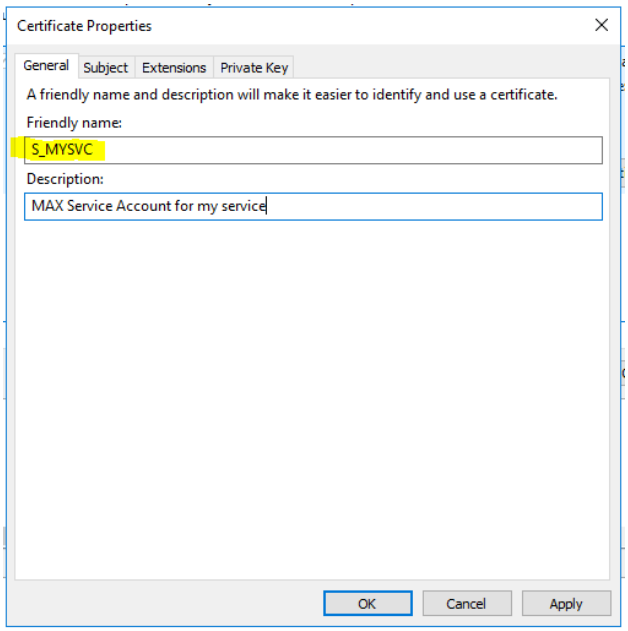
g. Set some options for the certificate. On the next screen, pick “Details”, then “Properties”:



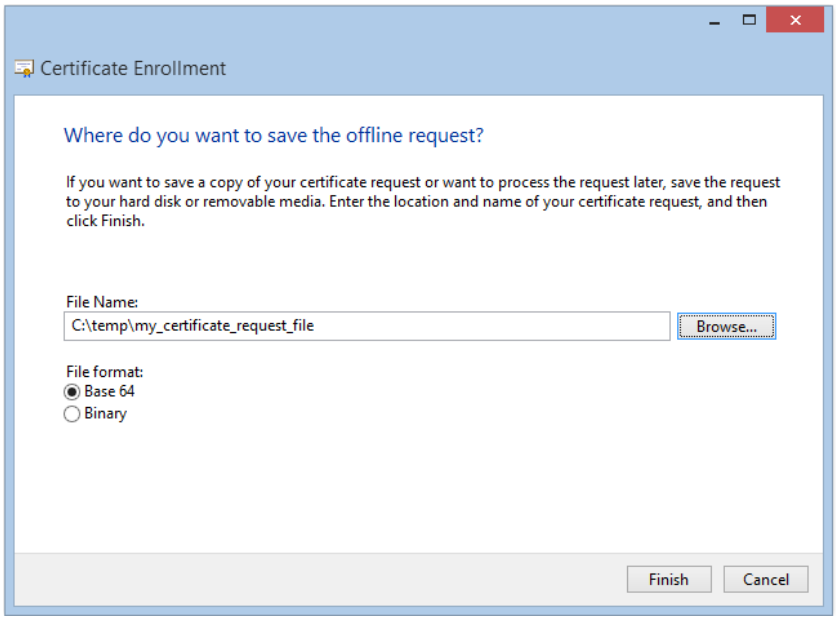
h. In the properties, you will set the following:
   i. Key size: 2048
   ii. Make the private key exportable
   iii. Hash algorithm: sha512



   iv. You can also set the "Friendly Name" in the "General" tab to make it easy to find your certificate later. **Helpful hint: use the MAX ID of the service account as the friendly name (e.g., S_MYSVC).**



i. Pick a place to store the CSR:



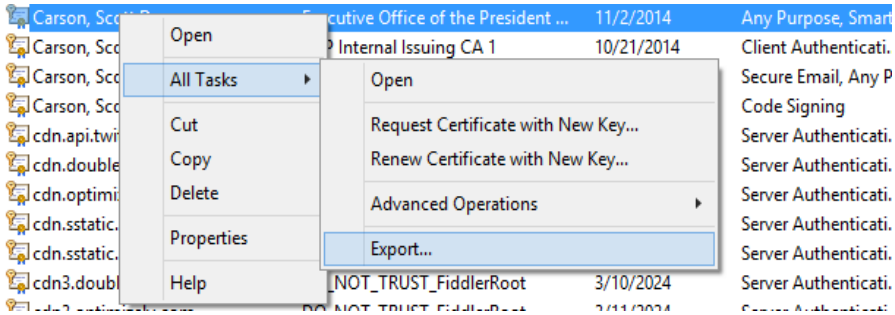
j. If you take a look at the result using notepad, you'll see something like this:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIICODCCAaECAQAwADCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAtr2qObIn
FYpfP2tY+Hot7p/vZHMLSZlRNJldmJQY+AdLulwj228G79TwAvaU78mmKql3a9xY
494icxEG1o3ouYpz04FnVoJkrdl9ZcJz7g+EcQ20tT6WXBT56VsobDDSmupoWWG9
bOwU284MkwCntC5kvyCf7eIJSt6AFpbF+UMCAwEAAaCB9zAaBgorBgEEAYI3DQID
MQwWCjYuMy45NjAwLjIwLgYJKoZIhvcNAQkOMSEwHzAdBgNVHQ4EFgQUCfOc7uFF
k36pBrzspH4q3Gi6hr8wQQYJKwYBBAGCNxUUMTQwMgIBBQwVRU9QUjExMDQyMC5E
Uy5FT1AuR09WDA1EU1xjYXJzb25fc2RhDAdNTUMuRVhFMGYGCisGAQQBgjcNAgIx
WDBWAgEAHk4ATQBpAGMAcgBvAHMAbwBmAHQAIABTAG8AZgB0AHcAYQByAGUAIABL
AGUAeQAgAFMAdABvAHIAYQBnAGUAIABQAHIAbwB2AGkAZABlAHIDAQAwDQYJKoZI
hvcNAQELBQADgYEAaI0xm9VwTkeNJfl1CJBgbJje+RPgTO1D1DTJoAZCL3moAxRi
mXpmigWNDeeWPjSj8nl8v6zutebnFsB8sAvgOxnkDDyK58HWUr3BraorcPJ61T5q
AIpttmMfx18bHPw8h0lfpbltR6uvYsGeSoW9g0ENKuF7jXNCRIruhj+9jNw=
-----END NEW CERTIFICATE REQUEST-----
```

k. Look in the "Certificate Enrollment Requests" folder in mmc, and you'll see your request, which you can recognize by its "friendly name".

2. Email the request to the MAX staff. You will receive a certificate in return.
3. Save this certificate to your system as a ".cer" file.
4. Import this certificate into your machine's certificate store. It should go in your "Personal" store.
   a. Right click on the "Certificates" object under "Personal", then select "All Tasks", "Import".
   b. Browse to the .cer file, and select it.
   c. Upon successful completion, you can verify that it worked as expected by right clicking on the certificate and selecting "All Tasks". One of the options should be "Manage Private Keys" – its presence indicates that this import did satisfy the request you created and that you do have the private key.
5. You may also receive a second certificate, the "Certification Authority" (CA) certificate. If you do, save this as a ".cer" file also, then import it into the "Trusted Root Certification Authorities" store

   .
6. Export the certificate (the one you imported in Step 4 above) by selecting it, then right-click, "All Tasks", "Export":



When you do, choose to export all the certificates in the trust chain, and – importantly – export the private key. You will be prompted to enter a password – do so and remember it. Save the certificate to a ".pfx" file.

7. Delete the certificate you just exported from the certificate store. The .pfx file is now your only copy of the certificate.

At this point, you have a complete MAX identity encapsulated in the .pfx file. Treat this file carefully as outlined below, since anyone who knows the password will be able to impersonate that identity.

Getting a client certificate on Linux

Creating a private key and signing request on Linux is considerably simpler, although the burden then falls on you to manage the resulting files. To create the request, use this command:

```
openssl req -newkey rsa:2048 -nodes -keyout S_MYSVC.key -out S_MYSVC.csr -subj
'/CN=request'
```

Then email the resulting request file (S_MYSVC.csr) to the MAX staff (be sure to let them know what MAX ID this is for). You'll get back a certificate file, which you can then save to S_MYSVC.crt. You'll then have the two pieces you need: the public certificate (in the example, S_MYSVC.crt) and the private key (in the example, S_MYSVC.key). *From this point on, please protect the private key file by making it readable only to you, as it's the equivalent of a password*. Anyone who knows your private key can log in to MAX as the service account using the .crt file and the .key file together.

Keeping and safeguarding your MAX certificate

Windows and Linux provide different mechanisms for limiting access to your private key. In Windows, keys are stored in a "key store". Each user has a separate key store, and the system has its own store as well. The key store provides a way to keep people from exporting the private key, and it provides an access control mechanism for limiting who can read it. In Linux, the private key is stored in a file, and normal file system permissions are used to regulate which accounts can read it.

Best practices for storing and using the certificate on Windows

The following are recommended practices for working with your MAX certificate:

1. Do not email the .pfx file, and do not post it on a web site.
2. Import the .pfx file into the certificate store *on the machine where you will use the certificate* (possibly different from the system where you just generated the certificate).
    a. Use mmc as above, but this time specify the "local machine" store instead of your personal store. You will have to do this with elevated permission.
    b. In the import process, **mark the private key as non-exportable** (important).
    c. Import the MAX CA (certificate authority) certificate that you imported in the previous section into the trusted root certificate authorities store on the target machine.
3. Remove the .pfx file from the machine where you will use the certificate.
4. Store the .pfx file in a place that is inaccessible to the system where you will use the certificate, preferably off-line.
5. Use a dedicated Windows service account to perform web requests using the certificate, and grant read access to the certificate's private key to that account.

Just to review, here's the situation:

1. The target system has your certificate and its private key. This certificate can be used to authenticate to MAX.
2. The target system won't export the private key, so someone who breaks into it (or stumbles upon it) can't take the certificate away and use it for something else.
3. Your .pfx file serves as a backup of your complete certificate (remember the password!) and is stored in a secure place, "away from the action".

Best practices for storing and using the certificate on Linux

As mentioned above, the private key (the .key file) is the piece of information that allows the service ID to log in to MAX. As such, you must protect it from unauthorized use. The following are recommended practices for working with the private key:

1. Back up your .key and .crt files somewhere that is offline; for example, on a CD or a thumb drive. If you need to keep your backup online, make sure that it isn't accessible via the web and that only you can access the files.
2. Make your private key file readable by the minimum number of parties required for your application to function. If your application runs as a particular Linux ID, then allow only that ID to access the private key file.
3. Do not email or post your private key file.
4. Do not use the same certificate on multiple machines (and in the process, do not copy the private key from machine to machine).

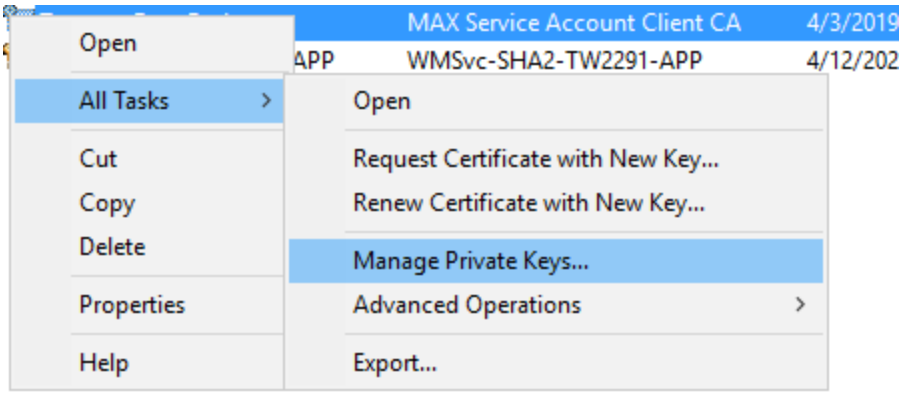
Working with your MAX Certificate

Once you have a MAX certificate, you can attach it to a web request for a MAX service. In doing so, the certificate will establish the identity of the web request as that of the service ID. The service ID functions just like any other MAX identity: you can grant permission for it to read and create pages in MAX Community, download and upload files, and so forth. You can also add it to MAX groups. Typically, you will use your certificate in non-interactive tasks such as scheduled tasks, cron jobs, or other scripts. You can also use your certificate from a program.

The techniques for using your certificate in a Windows or Linux script are very similar. In both cases, you will use the MAX.gov PIV authentication site (piv.max.gov) to validate the certificate, and request that it "hand off" your request to the site you're trying to access afterward.

Using your MAX certificate with Windows

A prerequisite to doing anything with the certificate is that you have to be able to read the private key. This means that if you're going to run a program interactively, you will need access. If you're running a script or a program as a scheduled task, the account under which that job runs must also have access to the private key. You can do this by opening the certificate store (again, using mmc), locating your certificate, right clicking, and selecting "Manage Private Keys", like this:



Then, add "Read" permission for the account in question, just as you would manage permissions on any other object.

Once you've made the private key accessible, you can use your certificate in a program (e.g., c# or vb) and/or in a PowerShell script; either way, you're using the same underlying framework. Here's an example of how to make a call to MAX using PowerShell:

First, open the local machine certificate store and select the certificate using the friendly name. Here's where you will use the fact that you provided the MAX ID as the friendly name when you requested the certificate:

```
$store = New-Object
System.Security.Cryptography.X509Certificates.X509Store("My","LocalMachine")

$store.Open("readonly")

$cert = $store.certificates | % { if ($_.FriendlyName -eq "S_MYSVC") { $_ }}
```

Then, you will "bounce" through the MAX PIV (certificate) authentication site with the certificate attached, specifying a MAX site to visit when authentication is complete. This example runs a simple MDS (Max Document Services) Excel report:

```
$download = invoke-webrequest  -Certificate $cert
'https://piv.max.gov/cas/login?service=https://mds.max.gov/get_spreadsheet?folder=test
s%26xls_filename=simple_user_test.xlsx'

[System.IO.File]::WriteAllText('simple_user_test.xlsx', $download.content)
```

Note: If your system uses proxy settings, you may have to specify them on the invoke-webrequest call.

Using your MAX certificate with Linux

Using the certificate with curl is the same idea as with PowerShell – you will create a web request and then "bounce" it through https://piv.max.gov with the certificate attached. The following command tells curl to follow the redirect from piv.max.gov back to the target MAX site (mds.max.gov), to attach the certificate to the request, and to maintain cookies along the way.

```
curl -L -j -b none --cert S_MYSVC.crt --key S_MYSVC.key
'https://piv.max.gov/cas/login?service=https://mds.max.gov/get_spreadsheet?folder=test
s%26xls_filename=simple_user_test.xlsx' >simple_user_test.xlsx
```

Just as in the Windows example, if your system requires proxy settings then you will have to specify them to curl.

Summary and further information

Now that you've read through this document, you know what a MAX certificate is, how to get one, and what to do with it. Hopefully you also come away with the sense that certificates are useful for a certain class of problem that you couldn't solve another way, but at the same time, a sense that using them requires care and diligence.

If you need help obtaining or using a MAX certificate, please get in contact with MAX Support at support@max.gov, by calling 202-395-6860 or by visiting https://support.max.gov.