



PNMsoft Knowledge Base

Sequence Administrator Guides

Configuring SharePoint 2013 with Remote Web Parts and ADFS

© 2016 PNMsoft All Rights Reserved

This document, including any supporting materials, is owned by PNMsoft Ltd and/or its affiliates and is for the sole use of the PNMsoft customers, PNMsoft official business partners, or other authorized recipients. This document may contain information that is confidential, proprietary or otherwise legally protected, and it may not be further copied, distributed or publicly displayed without the express written permission of PNMsoft Ltd. or its affiliates.

PNMsoft UK 38 Clarendon Road Watford Hertfordshire WD17 1JJ

Tel: +44(0)192 381 3420 • Email: info@pnmsoft.com • Website: www.pnmsoft.com

Microsoft Partner

Gold Application Development

TABLE OF CONTENTS

General Document Information	1
Purpose.....	1
Prerequisites.....	1
Overview	2
Installing and Configuring ADFS	2
Configuring SharePoint 2013	2
Configuring a SharePoint site to Use Claims	2
Configuring Flowtime web.config	3
Configuring Flowtime Site in IIS.....	7
Important Notes	7
Appendix A: PowerShell Scripts	8
PowerShell Script – Creating a New Trust Provider	8
PowerShell Script - Deleting the Trust Provider	9
Appendix B: Creating a 'Custom Claim Rule' in ADFS to Achieve a 'Domain\User' Authentication Type	10

General Document Information

Purpose

This document describes how to configure a SharePoint site with Remote Web Parts and with Claims Authentication using ADFS as STS and **SAML 1.1**.

Prerequisites

- You should be a Sequence and Server Administrator.
- Familiarity with Claims-based Authentication.
- Sequence v7.5 and above.
- SharePoint-less Flowtime installed.
- Remote Web Parts solution deployed on a SharePoint site.

Overview

The Flowtime end user environment runs on SharePoint. You can configure Flowtime to work with several types of authentication, including Claims-based authentication.

Two phases are required to configure Flowtime to work with Claims-based authentication:

1. Configuring SharePoint.
2. Configuring SharePoint-less Flowtime to use the claims that are provided by the infrastructure.

Installing and Configuring ADFS

Follow the instructions in this deployment guide:

<http://go.microsoft.com/fwlink/p/?LinkId=191723>

After installing ADFS, add a relying party (SharePoint 2013) according to:

<http://technet.microsoft.com/en-us/library/hh305235.aspx#relyparty>.

When configuring the claim rule, be aware that the claims provided here will be used by Flowtime to authenticate the user and at least one of the claims should contain a value that Flowtime can use (email, username, username + domain).

Export the token signing certificate from ADFS. This is described here:

<http://technet.microsoft.com/en-us/library/hh305235.aspx#ExportCert>

Copy the exported certificate file to the SharePoint deployment server in order to configure SharePoint to use this ADFS.

When adding a new 'Relying Party Trust', set the endpoint to be:

`https://<FLOWTIME_URL>/flowtime/_layouts/ProxyPage.aspx`

Set the identifier to be 'urn:splft' or come up with your own identifier and replace every entity in this guide that is named 'urn:splft' (audienceUri for example) with the desired identifier.

Configuring SharePoint 2013

Make sure that the SharePoint application has an SSL-enabled address.

Configure SharePoint according to the steps described here:

<http://technet.microsoft.com/en-us/library/hh305235.aspx#Phase3>.

The PowerShell scripts can be found in Appendix A.



Configuring a SharePoint site to Use Claims

Add the newly-trusted identity provider as an authentication provider for Flowtime as follows:

1. Open the SharePoint Central Administration.
2. Select **Manage Web Applications**.
3. Select the web application from the application list.
4. Select the **Authentication Providers** menu item.
5. Click the Default zone.
6. Check the **Trusted Identity Provider** checkbox.
7. Check the **Newly Created Trusted Identity Provider** checkbox.

- Click **OK** and wait for the dialog box to close.

In SharePoint 2013 Application Management, add User Policy to the users that will logon to the desired web application, as follows:

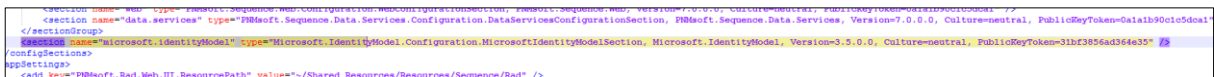
- Open SharePoint Central Administration.
- Select **Manage Web Applications**.
- Select the web application from the application list.
- Select the **User Policy** menu item.
- Click **Add Users**.
- Select the zone if applicable and click **Next**.
- To select the users that should have access to the application, click the  icon below the Users text area.
- In the **Find** textbox, enter the value of the identifying claim and click the  icon. Select the user from the list of users under the trusted identity provider name you created before.
- Click **Add** and then **OK**.
- Select the appropriate permissions and click **Finish**.
- Click **OK** on the *Policy for Web Application* dialog.

Configuring Flowtime web.config

Open the Flowtime web.config file and apply the following changes:

- Add the Microsoft IdentityModel configuration sections.
- Add the following to the `<configSections>` element

```
<section name="microsoft.identityModel"
type="Microsoft.IdentityModel.Configuration.MicrosoftIdentityModelSection, Microsoft.IdentityModel,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
```



```
<section name="web" type="Microsoft.Sequence.Web.Configuration.WebConfigurationSection, Microsoft.Sequence.Web, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a11b90c15dca1" />
</sectionGroup>
<!-- Add the following to the configSections element -->
<configSections>
<add name="microsoft.identityModel" type="Microsoft.IdentityModel.Configuration.MicrosoftIdentityModelSection, Microsoft.IdentityModel, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
</configSections>
appSettings>
<add key="PNMSOFT.Bad.Web.UI.ResourcePath" value="/Shared_Resources/Resources/Sequence/Bad" />
```

- Add the following to the `<assemblies>` element under the `<system.web>` `<compilation>` elements.

```
<add assembly="Microsoft.IdentityModel, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" />
```



```
</location>
<system.web>
<!--
Set compilation debug="true" to insert debugging
symbols into the compiled page. Because this
affects performance, set this value to true only
during development.
-->
<compilation defaultLanguage="c#" debug="true">
<assemblies>
<add assembly="System.Core, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089" />
<add assembly="System.Data.DataSetExtensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089" />
<add assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
<add assembly="System.Xml.Linq, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089" />
<add assembly="PNMSOFT.Sequence.Security, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a11b90c15dca1" />
<add assembly="PNMSOFT.Sequence, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a11b90c15dca1" />
<add assembly="PNMSOFT.Bad.Web.UI, Version=2012.03.1017.0, Culture=neutral, PublicKeyToken=29a01a93ac063d99" />
<add assembly="Microsoft.IdentityModel, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
</assemblies>
</buildProviders>
<add extension=".sgaux" type="PNMSOFT.Sequence.Web.Services.Compilation.WebServiceListenerBuildProvider, PNMSOFT.Sequence.Web.Services, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a11b90c15dca1" />
</buildProviders>
</compilation>
<!--
```

- Add the following to the `<system.webServer>` `<modules>` element.

```
<add name="WSFederationAuthenticationModule"
type="Microsoft.IdentityModel.Web.WSFederationAuthenticationModule, Microsoft.IdentityModel,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" preCondition="managedHandler" />
```

```
<add name="SessionAuthenticationModule" type="Microsoft.IdentityModel.Web.SessionAuthenticationModule,
Microsoft.IdentityModel, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
preCondition="managedHandler" />
```

```
<!--system.webServer-->
<validation validateIntegratedModeConfiguration="false" />
<modules>
<remove name="ScriptModule" />
<add name="ScriptModule" preCondition="managedHandler" type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
<add name="SequenceAuthenticationModule" type="PNMSoft.Sequence.Web.AuthenticationModule, PNMSoft.Sequence.Web, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1ab90c1c5dca1" />
<add name="SequenceModule" type="PNMSoft.Sequence.Web.NextFlowEngineHttpModule, PNMSoft.Sequence.Web, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1ab90c1c5dca1" />
<add name="SequenceFormModule" type="PNMSoft.Sequence.Form.Web.AuthenticationHttpModule, PNMSoft.Sequence.Form, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1ab90c1c5dca1" />
<add name="RadUploadModule" preCondition="integratedMode" type="PNMSoft.Rad.Web.UI.RadUploadHttpModule, PNMSoft.Rad.Web.UI, Version=2012.03.1017.0, Culture=neutral, PublicKeyToken=29a193ec043d92" />
<add name="FederatedAuthenticationModule" type="Microsoft.IdentityModel.Web.FederatedAuthenticationModule, Microsoft.IdentityModel, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" preCondition="managedHandler" />
<add name="SessionAuthenticationModule" type="Microsoft.IdentityModel.Web.SessionAuthenticationModule, Microsoft.IdentityModel, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" preCondition="managedHandler" />
</modules>
<handlers accessPolicy="Read, Script">
<remove name="WebServiceHandlerFactory-Integrated" />
<remove name="ScriptHandlerFactory" />
```

5. Add the following under the <configuration> element.

```
<microsoft.identityModel>
<service>
<audienceUri>
<add value="urn:splft" />
</audienceUri>
<federatedAuthentication>
<wsFederation passiveRedirectEnabled="true" issuer="https://sso.domain.local/adfs/ls/"
realm="urn:splft" requireHttps="true" persistentCookiesOnPassiveRedirects="true" />
<cookieHandler requireSsl="false" name="SPLTFedAuth" path="/" />
</federatedAuthentication>
<applicationService>
<claimTypeRequired>
</claimTypeRequired>
</applicationService>
<issuerNameRegistry
type="Microsoft.IdentityModel.Tokens.ConfigurationBasedIssuerNameRegistry, Microsoft.IdentityModel,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35">
<trustedIssuers>
<add thumbprint="f6 de 1d 44 6d a9 62 55 09 c6 48 46 66 a5 b6 b6 c6 62 fb 1b"
name="http://sso.domain.local/adfs/services/trust" />
</trustedIssuers>
</issuerNameRegistry>
<certificateValidation certificateValidationMode="None" />
</service>
</microsoft.identityModel>
```

```
</system.diagnostics>
<microsoft.identityModel>
<service>
<audienceUri>
<add value="urn:splft" />
</audienceUri>
<federatedAuthentication>
<wsFederation passiveRedirectEnabled="true" issuer="https://sso.domain.local/adfs/ls/" realm="urn:splft" requireHttps="true" persistentCookiesOnPassiveRedirects="true" />
<cookieHandler requireSsl="false" name="SPLTFedAuth" path="/" />
</federatedAuthentication>
<applicationService>
<claimTypeRequired>
</claimTypeRequired>
</applicationService>
<issuerNameRegistry type="Microsoft.IdentityModel.Tokens.ConfigurationBasedIssuerNameRegistry, Microsoft.IdentityModel, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35">
<trustedIssuers>
<add thumbprint="f6 de 1d 44 6d a9 62 55 09 c6 48 46 66 a5 b6 b6 c6 62 fb 1b" name="http://sso.domain.local/adfs/services/trust" />
</trustedIssuers>
</issuerNameRegistry>
<certificateValidation certificateValidationMode="None" />
</service>
</microsoft.identityModel>
</configuration>
```

6. If Flowtime is installed on the same machine and is using the same hostname, change the name attribute of the <cookieHandler> element to be other than "FedAuth".
7. Configure the <issuerNameRegistry> to match the organization's ADFS certificate settings.
8. Configure the realm and issuer attributes of the <wsFederatedAuthentication> section to match the organization's ADFS settings.
9. Configure the thumbprint attribute by using the thumbprint of the ADFS certificate that you have deployed to SharePoint.
10. Configure Sequence engine authentication to use claims-based authentication by overwriting the <authentication> node with the following:

```

    <authentication impersonate="false">
      <providers>
        <add type="PNMsoft.Sequence.Security.ClaimsIdentityAuthenticationProvider,
PNMsoft.Sequence.IdentityModel, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1" />
      </providers>
      <claims enabled="true">
        <IdentityClaims>
          <add
claimType="http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname"
originalIssuer="http://sso.domain.local/adfs/services/trust"

authenticationType="http://pnmsoft.com/sequence/2008/03/authentication/types/username" />
        </IdentityClaims>
      </claims>
    </authentication>

```

```

<sequence.engine>
  <authentication impersonate="false">
    <providers>
      <add type="PNMsoft.Sequence.Security.ClaimsIdentityAuthenticationProvider, PNMsoft.Sequence.IdentityModel, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1" />
    </providers>
    <claims enabled="true">
      <IdentityClaims>
        <add claimType="http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname" originalIssuer="http://sso.domain.local/adfs/services/trust"
authenticationType="http://pnmsoft.com/sequence/2008/03/authentication/types/username" />
      </IdentityClaims>
    </claims>
  </authentication>

```

The `claimType` attribute should match the claim type that holds the value used to authenticate the user against Flowtime.

The `authenticationType` attribute should correspond to the claim type value. The following options are available (Note: these are case sensitive!):

- usernameDomain (corresponds to `fldGroup` and `fldEmpUseName` in `tblEmployees`. Format should be `DOMAIN\User`)
- username (corresponds to `fldEmpUseName` in `tblEmployees`)
- email (corresponds to `fldEmail` in `tblEmployees`)

NOTE: in versions prior to Sequence 7.8, there was a spelling mistake in the `authenticationType`. So, if your version of Sequence is older than 7.8, change the `authenticationType` from `http://pnmsoft.com/sequence...` to: `http://pmnsoft.com/sequence...`

11. Modify the `<system.serviceModel>` section to match the following:

```

<system.serviceModel>
  <serviceHostingEnvironment aspNetCompatibilityEnabled="true" />
  <services>
    <service
name="PNMsoft.Sequence.Flowtime.Services.Messages.UserMessagesService">
      <endpoint address="" binding="webHttpBinding"
bindingConfiguration="webHttpBinding"
contract="PNMsoft.Sequence.Flowtime.Services.Messages.IUserMessagesService" />
    </service>
    <service
name="PNMsoft.Sequence.Flowtime.Services.Messages.GroupMessagesService">
      <endpoint address="" binding="webHttpBinding"
bindingConfiguration="webHttpBinding"
contract="PNMsoft.Sequence.Flowtime.Services.Messages.IGroupMessagesService" />
    </service>
    <service
name="PNMsoft.Sequence.Flowtime.Services.Instances.UserInstancesService">

```



```

        <endpoint address="" binding="webHttpBinding"
bindingConfiguration="webHttpBinding"
contract="PNMsoft.Sequence.Flowtime.Services.Instances.IUserInstancesService" />
    </service>
</service>
name="PNMsoft.Sequence.Flowtime.Services.Instances.ProcessInstancesService">
    <endpoint address="" binding="webHttpBinding"
bindingConfiguration="webHttpBinding"
contract="PNMsoft.Sequence.Flowtime.Services.Instances.IProcessInstancesService" />
    </service>
</service>
name="PNMsoft.Sequence.Flowtime.Services.Delegation.DelegationService">
    <endpoint address="" binding="webHttpBinding"
bindingConfiguration="webHttpBinding"
contract="PNMsoft.Sequence.Flowtime.Services.Delegation.IDelegationService" />
    </service>
</service>
name="PNMsoft.Sequence.Flowtime.Services.Delegators.DelegatorsService">
    <endpoint address="" binding="webHttpBinding"
bindingConfiguration="webHttpBinding"
contract="PNMsoft.Sequence.Flowtime.Services.Delegators.IDelegatorsService" />
    </service>
</service>
<service name="PNMsoft.Sequence.Flowtime.Services.UtilityService">
    <endpoint address="" binding="webHttpBinding"
bindingConfiguration="webHttpBinding" contract="PNMsoft.Sequence.Flowtime.Services.UtilityService"
/>
</service>
</services>
<bindings>
    <webHttpBinding>
        <binding name="webHttpBinding">
            <security mode="Transport">
                <transport clientCredentialType="None" />
            </security>
        </binding>
    </webHttpBinding>
</bindings>
</system.serviceModel>

```

```

<system.serviceModel>
  <serviceHostingEnvironment aspNetCompatibilityEnabled="true" />
  <services>
    <service name="PNMsoft.Sequence.Flowtime.Services.Messages.UserMessagesService">
      <endpoint address="" binding="webHttpBinding" bindingConfiguration="webHttpBinding" contract="PNMsoft.Sequence.Flowtime.Services.Messages.IUserMessagesService" />
    </service>
    <service name="PNMsoft.Sequence.Flowtime.Services.Messages.GroupMessagesService">
      <endpoint address="" binding="webHttpBinding" bindingConfiguration="webHttpBinding" contract="PNMsoft.Sequence.Flowtime.Services.Messages.IGroupMessagesService" />
    </service>
    <service name="PNMsoft.Sequence.Flowtime.Services.Instances.UserInstancesService">
      <endpoint address="" binding="webHttpBinding" bindingConfiguration="webHttpBinding" contract="PNMsoft.Sequence.Flowtime.Services.Instances.IUserInstancesService" />
    </service>
    <service name="PNMsoft.Sequence.Flowtime.Services.Instances.ProcessInstancesService">
      <endpoint address="" binding="webHttpBinding" bindingConfiguration="webHttpBinding" contract="PNMsoft.Sequence.Flowtime.Services.Instances.IProcessInstancesService" />
    </service>
    <service name="PNMsoft.Sequence.Flowtime.Services.Delegation.DelegationService">
      <endpoint address="" binding="webHttpBinding" bindingConfiguration="webHttpBinding" contract="PNMsoft.Sequence.Flowtime.Services.Delegation.IDelegationService" />
    </service>
    <service name="PNMsoft.Sequence.Flowtime.Services.Delegators.DelegatorsService">
      <endpoint address="" binding="webHttpBinding" bindingConfiguration="webHttpBinding" contract="PNMsoft.Sequence.Flowtime.Services.Delegators.IDelegatorsService" />
    </service>
    <service name="PNMsoft.Sequence.Flowtime.Services.UtilityService">
      <endpoint address="" binding="webHttpBinding" bindingConfiguration="webHttpBinding" contract="PNMsoft.Sequence.Flowtime.Services.UtilityService" />
    </service>
  </services>
  <bindings>
    <webHttpBinding>
      <binding name="webHttpBinding">
        <security mode="Transport">
          <transport clientCredentialType="None" />
        </security>
      </binding>
    </webHttpBinding>
  </bindings>
</system.serviceModel>

```

Configuring Flowtime Site in IIS

1. Open IIS and navigate to the Flowtime site. Expand it and expand the Flowtime application.
2. Navigate to `_layouts` and switch to 'Content View'.
3. Right-click `ProxyPage.aspx` and click **Switch to Features View**. Double-click **Authentication** and remove both 'Anonymous Authentication' and 'Windows Authentication'.
4. Repeat the previous step with the `RunTime.aspx` page.
5. In the Flowtime application, navigate to `_vti_bin/Flowtime` (Flowtime services folder) and double click **Authentication**. Remove 'Windows Authentication'.

Important Notes

- If you choose to use 'email' as the authentication type, it is very important to make sure that the users who are synchronized from your Active Directory to Sequence have a unique email address. Otherwise Sequence will choose the first user that appears in the 'employees table'.
If you have chosen the 'username' authentication type and you are syncing multiple domains with Sequence, make sure that all the users synced have a unique username. Otherwise Sequence will choose the first user that appears in the 'employees table'. Alternatively you can use the 'usernameDomain' authentication type (this will ensure that each user that is logged in, is logged in with a unique identifier). This will require you to assign a 'Custom Claim Rule' to the 'Relying Trust Party Claim Rules' (Appendix B includes an example of how to achieve this).
- Make sure the SharePoint Farm uses the same ADFS infrastructure and configuration as the Flowtime machine.
- Make sure to set the SharePoint Farm with the same `persistentCookiesOnPassiveRedirect` value as the Flowtime application. It is advisable that this property be set to `True`.

Appendix A: PowerShell Scripts

PowerShell Script – Creating a New Trust Provider

```
$ver = $host | select version
if($ver.Version.Major -gt 1) {
    if((Get-PSSnapin "Microsoft.Sharepoint.PowerShell" -ErrorAction
SilentlyContinue) -eq $null)
    {
        Add-PSSnapin "Microsoft.Sharepoint.PowerShell"
    }
}

#make sure that in your ADFS server the Relying Party Trust's "identifier" includes
"urn:sharepoint:sequence".
$realm = "urn:sharepoint:sequence"
$signInUrl = "<ADFS Signin URL>"
$providerName = "<Provider Name>"
$providerDescription = "<Provider Description>"

#import token signing certificate. Import any other certificates in the chain from
the token signing cert to the parent.
$tokenSignCertPath = "<Path for the certificate from ADFS>"
$tokenSignCert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2($tokenSignCertPath)
New-SPTTrustedRootAuthority -Name "Flowtime Token Signing Certificate" -Certificate
$tokenSignCert

# Claims mapping
$emailClaimMap = New-SPClaimTypeMapping -IncomingClaimType
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress" -
IncomingClaimTypeDisplayName "EmailAddress" -SameAsIncoming
$upnClaimMap = New-SPClaimTypeMapping -IncomingClaimType
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn" -
IncomingClaimTypeDisplayName "UPN" -SameAsIncoming
$roleClaimMap = New-SPClaimTypeMapping -IncomingClaimType
"http://schemas.microsoft.com/ws/2008/06/identity/claims/role" -
IncomingClaimTypeDisplayName "Role" -SameAsIncoming
$sidClaimMap = New-SPClaimTypeMapping -IncomingClaimType
"http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid" -
IncomingClaimTypeDisplayName "SID" -SameAsIncoming

#note that the Claim Type in the -IdentifierClaim attribute in the following command
will determine the type of claim that SharePoint will identify the user that logged
in.
$ap = New-SPTTrustedIdentityTokenIssuer -Name $providerName -Description
$providerDescription -Realm $realm -ImportTrustCertificate $tokenSignCert -
ClaimsMappings $emailClaimMap,$upnClaimMap,$roleClaimMap,$sidClaimMap -SignInUrl
$signInUrl -IdentifierClaim $emailClaimMap.InputClaimType
```

PowerShell Script - Deleting the Trust Provider

```
$ver = $host | select version
if($ver.Version.Major -gt 1) {
    if((Get-PSSnapin "Microsoft.Sharepoint.PowerShell" -ErrorAction
SilentlyContinue) -eq $null)
    {
        $canExecute = true
        Add-PSSnapin "Microsoft.Sharepoint.PowerShell"
    }
}

$tokenIssuerName = "<Provider Name>"
$certName = "Flowtime Token Signing Certificate"

if((Get-SPTrustedIdentityTokenIssuer -Identity $tokenIssuerName -ErrorAction
SilentlyContinue) -ne $null)
{
    Remove-SPTrustedIdentityTokenIssuer -Identity $tokenIssuerName -Confirm
}

#execute this code for each certificate in the certificate chain
if((Get-SPTrustedRootAuthority -Identity $certName -ErrorAction SilentlyContinue) -
ne $null)
{
    Remove-SPTrustedRootAuthority -Identity $certName -Confirm
}
```

Appendix B: Creating a 'Custom Claim Rule' in ADFS to Achieve a 'Domain\User' Authentication Type

Add the following rules in the following order:

1. This rule obtains the username and the UPN of a user.

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
```

```
=> add(store = "Active Directory", types = ("ssupn", "sswindowsaccountname"), query = ";userPrincipalName,sAMAccountName;{0}", param = c.Value);
```

2. This rule uses a RegEx to parse the UPN ([user@domain.local](#)) and return only the 'domain.local'

```
c:[Type == "ssupn"]
```

```
=> add(Type = "ssnewupn", Value = RegExReplace(c.Value, "^(.*)@", ""));
```

3. This rule issues a claim in the Domain\User format. In this example we used a RegEx to parse the 'domain.local' and return only 'domain'. You could alternatively hardcode the domain name or use a different RegEx according to your needs.

```
c1:[Type == "ssnewupn"]
```

```
&& c2:[Type == "sswindowsaccountname"]
```

```
=> issue(Type = "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Value = RegExReplace(c1.Value, "(\\..*$)", "") + "\\" + c2.Value);
```