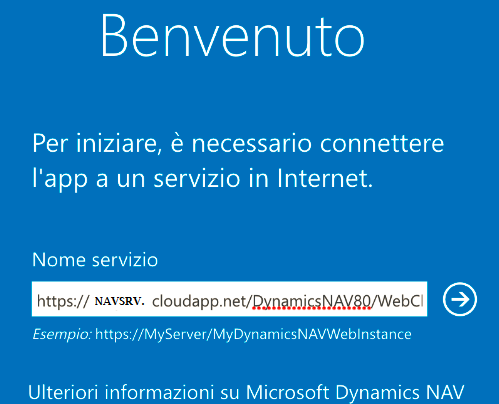
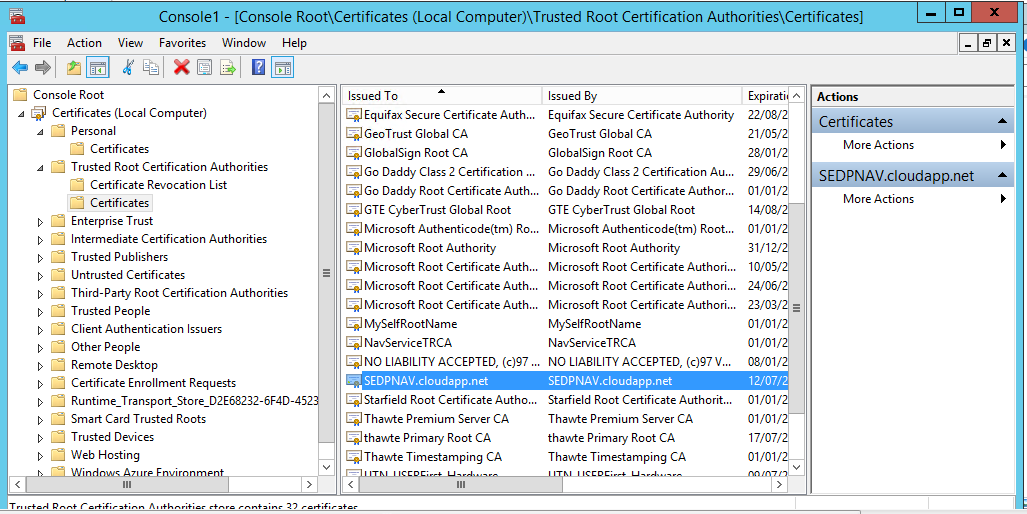
**Powershell Script to create a “Self-Signed Certificate” that works on Dynamics NAV Ipad APP**



**PURPOSE**  
**Automated scripts** (with Powershell) to **create a self-signed certificate** usable for Dynamics NAV App on Ipad (if you haven’t a Real Verified Certificate):  
  
**Powershell automation** based on procedure manual **"How to get Microsoft Dynamics NAV for tablets to connect using a self-signed certificate"**

[http://blogs.msdn.com/b/nav/archive/2014/10/09/how-to-g et-microsoft-dynamics-nav-for-tablets-to-connect-using-a-self-signed-certificate.aspx](http://blogs.msdn.com/b/nav/archive/2014/10/09/how-to-g%20et-microsoft-dynamics-nav-for-tablets-to-connect-using-a-self-signed-certificate.aspx)

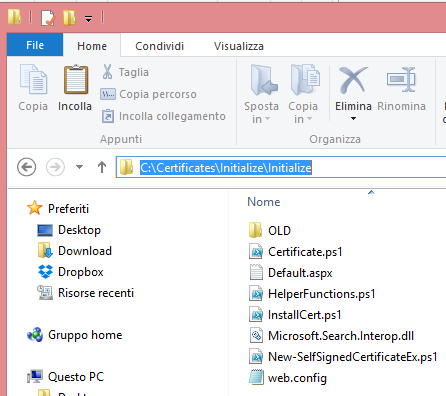
**STEP 1:**  
**REQUIREMENTS FOR USE PRE-USE CERTIFICATE "SELF-SIGNED"**  
- Best Practice Microsoft 🡪 use valid certificate purchased (eg: Verisight)  
- Suggested (but not mandatory) is to install at least the cumulative NAV Update 8 2015 (CU8), it needs to use better management of self-signed certificate (unofficial).  
  
**STEP 2:**  
**ELIMINATION OF CERTIIFICATES NOT ALREADY WORKING AND IMPORTED ONM IPAD**  
- First, if you have already created certificates for use with App and these don’t work, remove them all using the console:



**STEP 3:**  
**PROCEDURE FOR CREATING A NEW SELF-SIGNED CERTIFICATE**  
- This procedure must be used after the installation of Navision Server.  
  
**CREATION (BUILD) OF NEW CERTIFICATE**  
- This procedure creates / uses a new self-signed certificate  
- Unzip the file with everything you need (**KIT PRE-EXISTING available to Deploy NAV on AZURE**), put in the default example **"C:\Certificates\Initialize\Initialize\"**

**Initialize Standard.ZIP File**





Replace the existing file with this modified **PowerShell Script**:



Which in loop calls another file “**Certificate.ps1**”



This file **has been modified** to the case: “**Self-Signed Certificate tested on iPad App”** (was commented on what is not needed in our test case environment and changed the parameters)  
   
Open the file with **Powershell ISE as administrator** (**Run as administrator**) and check the settings and make changes if necessary.

**SCRIPT DETAIL** 🡪 **InstallCert.ps1**

# INITIALIZE CERTIFICATES FOLDER

$PSScriptRootV2 = "C:\Certificates\Initialize\Initialize\"

Set-StrictMode -Version 2.0

$verbosePreference = 'Continue'

$errorActionPreference = 'Stop'

# IMPORT MODULES & FUNCTIONS ON POWERSHELL

. (Join-Path $PSScriptRootV2 'HelperFunctions.ps1')

. ("c:\program files\Microsoft Dynamics NAV\80\Service\NavAdminTool.ps1")

Import-Module "C:\Users\Desktop\App e Certificati\485629\_ita\NAV.8.0.41779.IT.DVD\WindowsPowerShellScripts\Cloud\NAVAdministration\NAVAdministration.psm1"

. (Join-Path $PSScriptRootV2 'New-SelfSignedCertificateEx.ps1')

# CONFIG DEFAULT NAV SERVER AND USER (LOOP DO WHILE)

$CustomSettingsConfigFile = 'c:\program files\Microsoft Dynamics NAV\80\Service\CustomSettings.config'

$config = [xml](Get-Content $CustomSettingsConfigFile)

$serverInstance = $config.SelectSingleNode("//appSettings/add[@key='ServerInstance']").value

$NavServiceUser = Get-UserInput -Id NavAdminUser -Text "NAV Service Login" -Default "navsrv"

do

{

$err = $false

$CloudServiceName = Get-UserInput -Id CloudServiceName -Text "What is the name of your Cloud-Service" -Default "$env:COMPUTERNAME.cloudapp.net"

try

{

$myIP = Get-MyIp

$dnsrecord = Resolve-DnsName $CloudServiceName -ErrorAction SilentlyContinue -Type A

if (!($dnsrecord) -or ($dnsrecord.Type -ne "A") -or ($dnsrecord.IPAddress -ne $myIP)) {

Write-Host -ForegroundColor Red "That is NOT your Cloud Service Name (Did you name your Cloud Service something different from the Virtual Machine?)"

Write-Host -ForegroundColor Red "Please find the correct Cloud Service Name in the Azure Management Portal."

$err = $true

}

}

catch {}

} while ($err)

# CREATE NEW INETPUB HTTP DIRECTORY > MAP ON NEW CERTIFICATE (OR ON EXISTING)

$httpWebSiteDirectory = "C:\inetpub\wwwroot\http"

new-item -Path $httpWebSiteDirectory -ItemType Directory -Force

. (Join-Path $PSScriptRootV2 'Certificate.ps1')

# GRANTE ACCESS TO CERTIFICATE TO USER RUNNING SERVICE TIER

Grant-AccountAccessToCertificatePrivateKey -CertificateThumbprint $thumbprint -ServiceAccountName $NavServiceUser

# CHANGE CONFIGURATION

Set-NAVServerConfiguration $serverInstance -KeyName "ServicesCertificateThumbprint" -KeyValue $thumbprint

Set-NAVServerConfiguration $serverInstance -KeyName "PublicWebBaseUrl" -KeyValue ('https://' + $PublicMachineName + '/' + $serverInstance + '/WebClient/')

#Set-NAVServerConfiguration $serverInstance -KeyName "SOAPServicesSSLEnabled" -KeyValue 'true'

#Set-NAVServerConfiguration $serverInstance -KeyName "SOAPServicesEnabled" -KeyValue 'true'

#Set-NAVServerConfiguration $serverInstance -KeyName "ODataServicesSSLEnabled" -KeyValue 'true'

#Set-NAVServerConfiguration $serverInstance -KeyName "ODataServicesEnabled" -KeyValue 'true'

#Set-NAVServerConfiguration $serverInstance -KeyName "PublicODataBaseUrl" -KeyValue ('https://' +$PublicMachineName + ':7048/' + $serverInstance + '/OData/')

#Set-NAVServerConfiguration $serverInstance -KeyName "PublicSOAPBaseUrl" -KeyValue ('https://' + $PublicMachineName + ':7047/' + $serverInstance + '/WS/')

#Set-NAVServerConfiguration $serverInstance -KeyName "PublicWinBaseUrl" -KeyValue ('DynamicsNAV://' + $PublicMachineName + ':7046/' + $serverInstance + '/')

#Set-NAVServerConfiguration $serverInstance -KeyName "ClientServicesCredentialType" -KeyValue "Windows"

#Set-NAVServerConfiguration $serverInstance -KeyName "ServicesDefaultCompany" -KeyValue $Company

# RESTART NAV SERVICE TIER

Set-NAVServerInstance -ServerInstance $serverInstance -Restart

# ADD FIREWALL RUKES FOR SOAP AND OData

netsh advfirewall firewall add rule name="Microsoft Dynamics NAV Web Client SSL" dir=in action=allow protocol=tcp localport=443 remoteport=any

#netsh advfirewall firewall add rule name="Microsoft Dynamics NAV SOAP Services" dir=in action=allow protocol=tcp localport=7047 remoteport=any

#netsh advfirewall firewall add rule name="Microsoft Dynamics NAV OData Services" dir=in action=allow protocol=tcp localport=7048 remoteport=any

# REMOVE THE DEFAULT IIS WEBSITE

Remove-DefaultWebSite -ErrorAction SilentlyContinue

# REMOVE BINDING FOR WEB CLIENT

Get-WebBinding -Name "Microsoft Dynamics NAV 2015 Web Client" | Remove-WebBinding

# ADD SSL BINDING TO WEB CLIENT (ADD thumbprint KEY)

New-SSLWebBinding -Name "Microsoft Dynamics NAV 2015 Web Client" -Thumbprint $thumbprint

# CREATE NEW HTTO SITE

if (!(Get-Website -Name http)) {

# Create the web site

Write-Verbose "Creating Web Site"

New-Website -Name http -IPAddress \* -Port 80 -PhysicalPath $httpWebSiteDirectory -Force

}

Copy-Item (Join-Path $PSScriptRootV2 'Default.aspx') "$httpWebSiteDirectory\Default.aspx"

Copy-Item (Join-Path $PSScriptRootV2 'web.config') "$httpWebSiteDirectory\web.config"

# ADD NEW FIREWALL RULE : PORT 80

Write-Verbose "Opening Firewall"

New-FirewallPortAllowRule -RuleName "HTTP access" -Port 80

# CHANGE "Web.config" File

$WebConfigFile = "C:\inetpub\wwwroot\$serverInstance\Web.config"

$WebConfig = [xml](Get-Content $WebConfigFile)

$WebConfig.SelectSingleNode("//configuration/DynamicsNAVSettings/add[@key='DnsIdentity']").value=$dnsidentity

$WebConfig.SelectSingleNode("//configuration/DynamicsNAVSettings/add[@key='ClientServicesCredentialType']").value="Windows"

$WebConfig.Save($WebConfigFile)

#$WebConfig.SelectSingleNode("//configuration/DynamicsNAVSettings/add[@key='Company']").value=$Company

#$WebConfig.SelectSingleNode("//configuration/DynamicsNAVSettings/add[@key='HelpServer']").value="$PublicMachineName"

# TURN OFF IE ENANCHED SECURITY MODE (For Testing IE on server)

Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed Components\{A509B1A7-37EF-4b3f-8CFC-4F3A74704073}" -Name "IsInstalled" -Value 0

Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed Components\{A509B1A8-37EF-4b3f-8CFC-4F3A74704073}" -Name "IsInstalled" -Value 0

**SCRIPT DETAIL 🡪** “Certificate.Ps1”

# CERTIFICATE MANAGING - "SELF SIGNED" OR VERIFIED

$certificatePfxFile = Get-UserInput -Id CertificatePfxFile -Text "Certificate Pfx File (Empty for using Self Signed Certificate)"

$selfsigned = (!$certificatePfxFile)

if ($selfsigned) {

$certificatePfxFile = Join-Path $PSScriptRootV2 'Certificate.pfx'

$certificatePfxPassword = 'P@ssword1'

if (!(Test-Path $certificatePfxFile)) {

New-SelfSignedCertificateEx -Subject "CN=$CloudServiceName" -IsCA $true -Exportable -Path $certificatePfxFile -Password (ConvertTo-SecureString -String $certificatePfxPassword -AsPlainText -Force)

}

$cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($certificatePfxFile, $certificatePfxPassword)

$CertificateCerFile = (Join-Path $PSScriptRootV2 "$CloudServiceName.cer")

Export-Certificate -Cert $cert -FilePath $CertificateCerFile

Copy-Item $CertificateCerFile -Destination "C:\Users\Public\Desktop\$CloudServiceName.cer"

Copy-Item $CertificateCerFile -Destination "C:\inetpub\wwwroot\http\Certificate.cer"

$thumbprint = $cert.Thumbprint

if (!(Get-Item Cert:\LocalMachine\my\$thumbprint -ErrorAction SilentlyContinue)) {

Import-PfxFile -PfxFile $certificatePfxFile -PfxPassword (ConvertTo-SecureString -String $certificatePfxPassword -AsPlainText -Force)

}

} else {

$certificatePfxPassword = Get-UserInput -Id CertificatePfxPassword -Text "Certificate Pfx Password"

# Import certificate

$cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($certificatePfxFile, $certificatePfxPassword)

$thumbprint = $cert.Thumbprint

if (!(Get-Item Cert:\LocalMachine\my\$thumbprint -ErrorAction SilentlyContinue)) {

Import-PfxCertificate -FilePath $certificatePfxFile -CertStoreLocation cert:\localMachine\my -Password (ConvertTo-SecureString -String $certificatePfxPassword -AsPlainText -Force)

$certistrusted = $true

try {

$certistrusted = Test-Certificate –Cert "cert:\currentuser\my\$thumbprint" -Policy SSL

} catch {

$certistrusted = $false

}

if (!$certistrusted) {

# Self signed certificate created on another machine (for load balancing purposes)

Import-PfxCertificate -FilePath $certificatePfxFile -CertStoreLocation cert:\localMachine\root -Password (ConvertTo-SecureString -String $certificatePfxPassword -AsPlainText -Force)

}

}

}

$dnsidentity = $cert.GetNameInfo('SimpleName',$false)

if ($dnsidentity.StartsWith('\*')) {

$dnsidentity = $dnsidentity.Substring($dnsidentity.IndexOf('.')+1)

}

if ($selfsigned) {

$PublicMachineName = $CloudServiceName

} else {

# Public DNS name

$PublicMachineName = ($CloudServiceName.Split('.')[0] + ".$dnsidentity")

$PublicMachineName = Get-UserInput -Id PublicMachineName -Text "What DNS name points to your service" -Default $PublicMachineName

if ($PublicMachineName -ne $CloudServiceName) {

$dnsrecord = Resolve-DnsName $PublicMachineName -ErrorAction SilentlyContinue -Type CNAME

if (!($dnsrecord) -or ($dnsrecord.Type -ne "CNAME") -or ($dnsrecord.NameHost -ne $CloudServiceName)) {

Write-Host -ForegroundColor Red "You need to create a CNAME record for $PublicMachineName that points to $CloudServiceName"

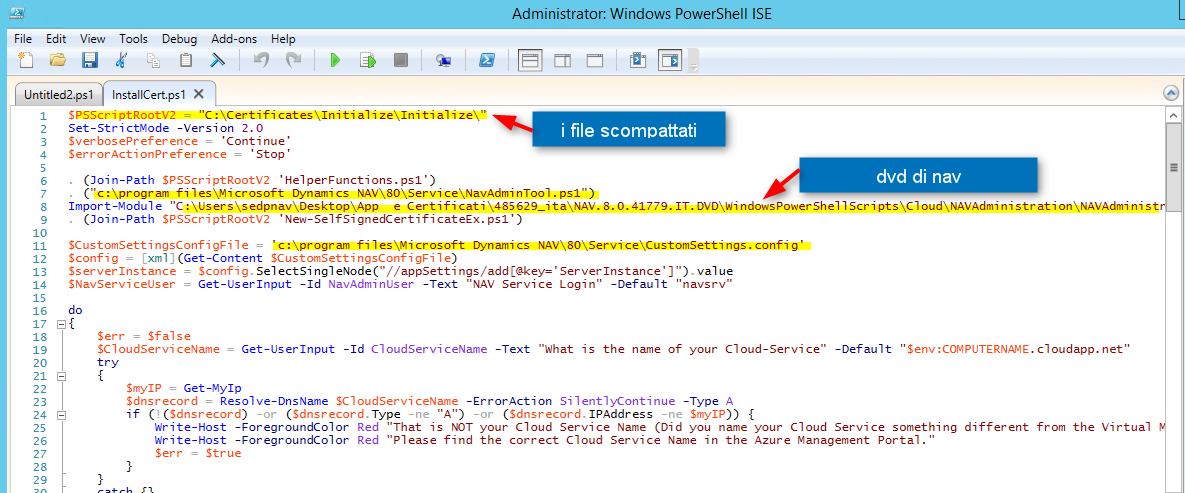
Start-Sleep -Seconds 30

}

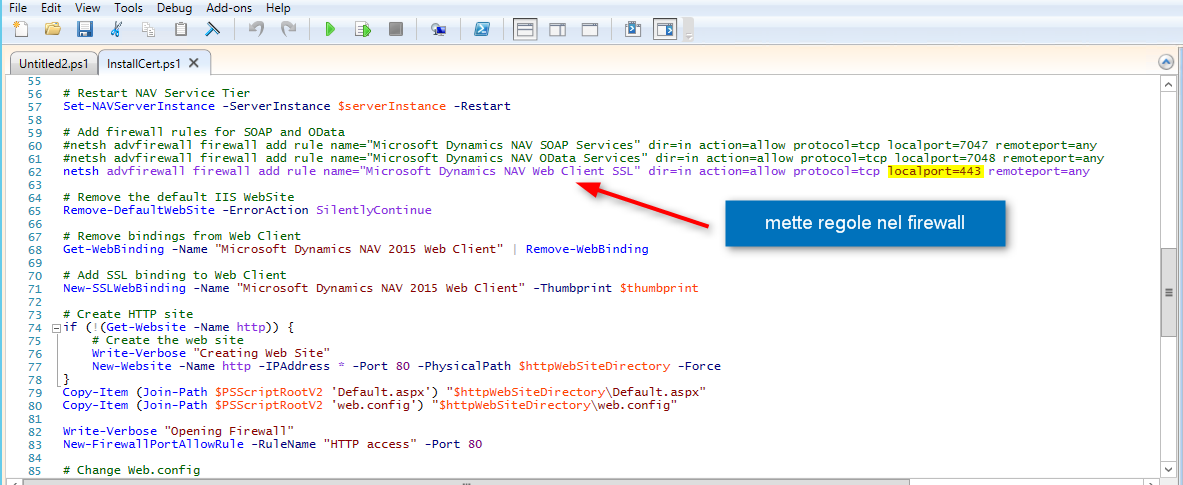
}

}

**“POWERSHELL ISE” SAMPLES SCREENS**





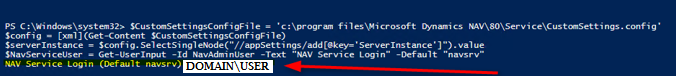


**STEP 4:**

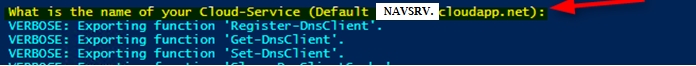
**Running the file "InstallCert.ps1"**

- From line to line with **F8 (debug) in powershell ISE** so if there are errors you can understand and correct, **or run the entire script with F5 if you are unsure**.

**STEP 5:  
INPUT PARAMETERS**  
**Two parameters** are requested:  
- The **first** (Default Login)



-The **second**: (NAV Server: press enter taking one proposed)



**The script stops and restarts the services,** once finished, **take the certificate and send it on iPad** (example 🡪 send to email)  
  
**STEP 6:**  
**INSTALLATION CERTIFICATE ON IPAD**  
-Click on the file in the mail and **follow the installation procedure on Ipad**.  
  
**STEP 7:**  
**LAUNCH DYNAMICS NAV APP**  
-Finished open "Dynamics NAV App" and enter the address

**“To Run” Dynamics NAV Ipad App**

* Download Ipad App : Dynamics NAV from App Store
* Install Certificate on Ipad (Double Click on .CER File)
* Run Ipad App and set **NAV Server Link** : **EXAMPLE** <https://navserver.cloudapp.net/DynamicsNAV80/WebClient>
* Insert username e password User: NAV (DOMAIN\User) PWD: Password

BINGO !

Have a nice Day !