# How to request a Token from Azure ACS via the OAuth v2 Protocol (CS\VBAzureACSWithOauth)

## Introduction

## When your web applications and services handle authentication using ACS, the client must obtain a security token issued by ACS to log in to your application or service. In order to obtain this ACS-issued token (output token), the client must either authenticate directly with ACS or send ACS a security token issued by its identity provider (input token). ACS validates this input security token, processes the identity claims in this token through the ACS rules engine, calculates the output identity claims, and issues an output security token.

## This topic describes the methods of requesting a token from ACS via the OAuth V2 protocol. All token requests via the OAuth V2protocol are transmitted over SSL. ACS always issues a Simple Web Token (SWT) via the OAuth V2 protocol, in response to a correctly formatted token request. All token requests via the OAuth V2protocol are sent to ACS in an HTTP POST. You can request an ACS token via the OAuth V2 protocol from any platform that can make an HTTPS FORM POST: .NET Framework, Windows Communication Foundation (WCF), Silverlight, ASP.NET, Java, Python, Ruby, PHP, Flash, and other platforms.

## Building the Sample

First change all the parameters to yours.

Then you can run this application directly.

## Running the Sample

The code below shows how to generate an SWT token.

|  |
| --- |
| -Code block start-  --C# code snippet start--  namespace CSAzureACSWithOauth  {  public class TokenFactory  {  string signingKey;  string issuer;  public TokenFactory(string issuer, string signingKey)  {  this.issuer = issuer;  this.signingKey = signingKey;  }  public string CreateToken()  {  StringBuilder builder = new StringBuilder();  // add the issuer name  builder.Append("Issuer=");  builder.Append(HttpUtility.UrlEncode(this.issuer));  string signature = this.GenerateSignature(builder.ToString(), this.signingKey);  builder.Append("&HMACSHA256=");  builder.Append(signature);  return builder.ToString();  }  private string GenerateSignature(string unsignedToken, string signingKey)  {  HMACSHA256 hmac = new HMACSHA256(Convert.FromBase64String(signingKey));  byte[] locallyGeneratedSignatureInBytes = hmac.ComputeHash(Encoding.ASCII.GetBytes(unsignedToken));  string locallyGeneratedSignature = HttpUtility.UrlEncode(Convert.ToBase64String(locallyGeneratedSignatureInBytes));  return locallyGeneratedSignature;  }  }  }  --C# code snippet end--  --VB code snippet start--  Public Class TokenFactory  Private signingKey As String  Private issuer As String  Public Sub New(issuer As String, signingKey As String)  Me.issuer = issuer  Me.signingKey = signingKey  End Sub  Public Function CreateToken() As String  Dim builder As New StringBuilder()  ' add the issuer name  builder.Append("Issuer=")  builder.Append(HttpUtility.UrlEncode(Me.issuer))  Dim signature As String = Me.GenerateSignature(builder.ToString(), Me.signingKey)  builder.Append("&HMACSHA256=")  builder.Append(signature)  Return builder.ToString()  End Function  Private Function GenerateSignature(unsignedToken As String, signingKey As String) As String  Dim hmac As New HMACSHA256(Convert.FromBase64String(signingKey))  Dim locallyGeneratedSignatureInBytes As Byte() = hmac.ComputeHash(Encoding.ASCII.GetBytes(unsignedToken))  Dim locallyGeneratedSignature As String = HttpUtility.UrlEncode(Convert.ToBase64String(locallyGeneratedSignatureInBytes))  Return locallyGeneratedSignature  End Function  End Class  --VB code snippet end--  -Code block end- |

The code below shows how to request a token from ACS via the OAuth V2 protocol.

|  |
| --- |
| -Code block start-  --C# code snippet start--  class Program  {  public const string ACSNameSpace = "{Your-ACS-NameSpace}";  static void Main(string[] args)  {  var accessToken = HttpUtility.UrlDecode(GetTokenBySymmetricKey("http://dinohy.com/"));  Console.WriteLine(accessToken);  Console.ReadLine();  }  public static string GetTokenByPassword(string scope)  {  try  {  const string identityName = "{Service-Identity-Name}";  const string identityPassword = "{Password}";  // Request a token from ACS  var client = new WebClient();  var address = new Uri(string.Format("https://{0}.accesscontrol.windows.net/v2/OAuth2-13", ACSNameSpace));  var values = new NameValueCollection();  values.Add("grant\_type", "client\_credentials");  values.Add("client\_id", identityName);  values.Add("client\_secret", identityPassword);  values.Add("scope", scope);  byte[] responseBytes = client.UploadValues(address, "POST", values);  string response = Encoding.UTF8.GetString(responseBytes);  // Parse the JSON response and return the access token  var serializer = new JavaScriptSerializer();  var decodedDictionary = serializer.DeserializeObject(response) as Dictionary<string, object>;  return decodedDictionary["access\_token"] as string;  }  catch (WebException wex)  {  string value = new StreamReader(wex.Response.GetResponseStream()).ReadToEnd();  throw;  }  }  public static string GetTokenBySymmetricKey(string scope)  {  try  {  // Request a token from ACS  var client = new WebClient();  var address = new Uri(string.Format("https://{0}.accesscontrol.windows.net/v2/OAuth2-13", ACSNameSpace));  var values = new NameValueCollection();  values.Add("grant\_type", "<http://schemas.xmlsoap.org/ws/2009/11/swt-token-profile-1.0>");  values.Add("assertion", createSWT("{Service-Identity-Name}", "0ytBPxRB6nc05zv6mjP2aK8rCWWPnP3fR+IDTDHEfSM="));  values.Add("scope", scope);  byte[] responseBytes = client.UploadValues(address, "POST", values);  string response = Encoding.UTF8.GetString(responseBytes);  // Parse the JSON response and return the access token  var serializer = new JavaScriptSerializer();  var decodedDictionary = serializer.DeserializeObject(response) as Dictionary<string, object>;  return decodedDictionary["access\_token"] as string;  }  catch (WebException wex)  {  string value = new StreamReader(wex.Response.GetResponseStream()).ReadToEnd();  throw;  }  }  public static string GetTokenFromAcsBySAML(string scope, string samlToken)  {  //For how to create a samlToken please refer to:  //http://msdn.microsoft.com/en-us/library/aa355062(v=vs.110).aspx  try  {  // Request a token from ACS  var client = new WebClient();  var address = new Uri(string.Format("https://{0}.accesscontrol.windows.net/v2/OAuth2-13",ACSNameSpace));  var values = new NameValueCollection();  values.Add("grant\_type", "saml2-bearer");  values.Add("assertion", samlToken);  values.Add("scope", scope);  byte[] responseBytes = client.UploadValues(address, "POST", values);  string response = Encoding.UTF8.GetString(responseBytes);  // Parse the JSON response and return the access token  var serializer = new JavaScriptSerializer();  var decodedDictionary = serializer.DeserializeObject(response) as Dictionary<string, object>;  return decodedDictionary["access\_token"] as string;  }  catch (WebException wex)  {  string value = new StreamReader(wex.Response.GetResponseStream()).ReadToEnd();  throw;  }  }  public static string createSWT(string issuer, string signingKey)  {  var factory = new TokenFactory(issuer, signingKey);  return factory.CreateToken();  }  }  --C# code snippet end--  --VB code snippet start--  Module Module1  Public Const ACSNameSpace As String = "{Your-ACS-NameSpace}"  Sub Main()  Dim accessToken = HttpUtility.UrlDecode(GetTokenBySymmetricKey("http://dinohy.com/"))  Console.WriteLine(accessToken)  Console.ReadLine()  End Sub  Public Function GetTokenByPassword(scope As String) As String  Try  Const identityName As String = "{Service-Identity-Name}"  Const identityPassword As String = "{Password}"  ' Request a token from ACS  Dim client = New WebClient()  Dim address = New Uri(String.Format("https://{0}.accesscontrol.windows.net/v2/OAuth2-13", ACSNameSpace))  Dim values = New NameValueCollection()  values.Add("grant\_type", "client\_credentials")  values.Add("client\_id", identityName)  values.Add("client\_secret", identityPassword)  values.Add("scope", scope)  Dim responseBytes As Byte() = client.UploadValues(address, "POST", values)  Dim response As String = Encoding.UTF8.GetString(responseBytes)  ' Parse the JSON response and return the access token  Dim serializer = New JavaScriptSerializer()  Dim decodedDictionary = TryCast(serializer.DeserializeObject(response), Dictionary(Of String, Object))  Return TryCast(decodedDictionary("access\_token"), String)  Catch wex As WebException  Dim value As String = New StreamReader(wex.Response.GetResponseStream()).ReadToEnd()  Throw  End Try  End Function  Public Function GetTokenBySymmetricKey(scope As String) As String  Try  ' Request a token from ACS  Dim client = New WebClient()  Dim address = New Uri(String.Format("https://{0}.accesscontrol.windows.net/v2/OAuth2-13", ACSNameSpace))  Dim values = New NameValueCollection()  values.Add("grant\_type", "<http://schemas.xmlsoap.org/ws/2009/11/swt-token-profile-1.0>")  values.Add("assertion", createSWT("{Service-Identity-Name}", "0ytBPxRB6nc05zv6mjP2aK8rCWWPnP3fR+IDTDHEfSM="))  values.Add("scope", scope)  Dim responseBytes As Byte() = client.UploadValues(address, "POST", values)  Dim response As String = Encoding.UTF8.GetString(responseBytes)  ' Parse the JSON response and return the access token  Dim serializer = New JavaScriptSerializer()  Dim decodedDictionary = TryCast(serializer.DeserializeObject(response), Dictionary(Of String, Object))  Return TryCast(decodedDictionary("access\_token"), String)  Catch wex As WebException  Dim value As String = New StreamReader(wex.Response.GetResponseStream()).ReadToEnd()  Throw  End Try  End Function  Public Function GetTokenFromAcsBySAML(scope As String, samlToken As String) As String  'For how to create a samlToken please refer to:  '<http://msdn.microsoft.com/en-us/library/aa355062(v=vs.110).aspx>  Try  ' Request a token from ACS  Dim client = New WebClient()  Dim address = New Uri(String.Format("https://{0}.accesscontrol.windows.net/v2/OAuth2-13", ACSNameSpace))  Dim values = New NameValueCollection()  values.Add("grant\_type", "saml2-bearer")  values.Add("assertion", samlToken)  values.Add("scope", scope)  Dim responseBytes As Byte() = client.UploadValues(address, "POST", values)  Dim response As String = Encoding.UTF8.GetString(responseBytes)  ' Parse the JSON response and return the access token  Dim serializer = New JavaScriptSerializer()  Dim decodedDictionary = TryCast(serializer.DeserializeObject(response), Dictionary(Of String, Object))  Return TryCast(decodedDictionary("access\_token"), String)  Catch wex As WebException  Dim value As String = New StreamReader(wex.Response.GetResponseStream()).ReadToEnd()  Throw  End Try  End Function  Public Function createSWT(issuer As String, signingKey As String) As String  Dim factory = New TokenFactory(issuer, signingKey)  Return factory.CreateToken()  End Function  End Module  --VB code snippet end--  -Code block end- |

## More Information

MSDN:

http://msdn.microsoft.com/en-us/library/hh674475.aspx