**SelfIssuedTokenSubmission**

This application demonstrates how the Generic Worker can be used with self-issued authentication tokens. When an organization already has some sort of authentication system in their servers, it should not be necessary to authenticate to the Generic Worker by providing username and password. Normal authentication with username and password is already demonstrated in most of the demo applications.

When the user supplies its username and password, this login information is checked against the Azure storage and an authentication token is issued, when the credentials are correct. This token contains Claims and Roles that authorizes the user to take actions.

This token can also be created programmatically, if the user already is authenticated in the users system. The code below shows how these Claim list is constructed and the role of researcher is assigned to the self-issued token. The commented-out code shows how to add the role of administrator.

claims.Add(new Claim(ClaimTypes.Name, "Name Joe", Rights.PossessProperty));

claims.Add(new Claim(ClaimTypes.GivenName, "Joe", Rights.PossessProperty));

claims.Add(new Claim(ClaimTypes.Surname, "Bloggs", Rights.PossessProperty));

claims.Add(new Claim(ClaimTypes.Email, "joe@university.edu", Rights.PossessProperty));

//claims.Add(new Claim("http://schemas.microsoft.com/ws/2008/06/identity/claims/role", "VENUS-C Compute Administrator", Rights.PossessProperty));

claims.Add(new Claim("http://schemas.microsoft.com/ws/2008/06/identity/claims/role", "VENUS-C Researcher", Rights.PossessProperty));

A different version of the secure client creation method is then called to supply the constructed claim list and to submit with the self-issued token.

// create secure client

var jobSubmissionPortal = GenericWorkerJobManagementClient.CreateSecureClient(submissionEndpoint, claimSet, clientCert, serviceCert);

To configure this application to run correctly, the thumbprint of the certificate must be replaced with "‎<YOUR CERTIFICATE THUMBPRINT UPPERCASE>" in the code below.

var serviceCert = LookupCertificate2(

StoreLocation.LocalMachine, StoreName.My.ToString(),

"‎<YOUR CERTIFICATE THUMBPRINT UPPERCASE>", X509FindType.FindByThumbprint);

The CN Name of the certificate must be replaced with "Windows Azure Tools" in the following line.

var submissionEndpoint = new EndpointAddress(new Uri(secureGenericWorkerURL), new DnsEndpointIdentity("Windows Azure Tools"));

The secureGenericWorkerUrl must be assigned to the Secure Service Url. If the application executes correctly, below is the console screen you should see.

