Alachisoft®

NCache™
Distributed Cache for .NET

Getting Started with NCache for Azure Guide

August 12, 2014

Contents

1. NCache for Azure Clients in Microsoft Azure

1.1. Overview

Microsoft Azure is an open and flexible cloud platform that enables quick development, deployment and management of applications across global network of Microsoft datacenters.

NCachefor Azure is an extremely fast and scalable in-memory distributed cache that caches application data and reduces expensive database trips. NCache for Azure in Microsoft Azure Virtual Machines (VMs) is utilized in the client applications using NCache for Azure remote client libraries. NCache for Azure remote clients are theclient applications running on different VMs than the cache cluster.

1.2. NCache for Azure Remote Clients

These NCache for Azure remote clients on Microsoft Azure can be the .NET applications running remotely as Azure Websites, Web Roles, Worker Roles or Virtual Machines in Microsoft Azure.

1.2.1. Client Applications as Azure Virtual Machines

You can choose to deploy your .NET applications directly on Microsoft Azure VMs. In this case, you have the option of either deploying NCache for Azure client libraries with the application or installing an NCache for Azure remote client on this Virtual Machine through a Windows Installer .msi file.

1.2.2. Client Applications as Web Roles or Worker Roles

You can choose to deploy your .NET application as a Web Role or Worker Role. NCache for Azure client libraries are deployed along with your application. NCache for Azure also provides an InProc or OutProc client cache in this environment.

2. NCache for Azure Clients as Azure Virtual Machines

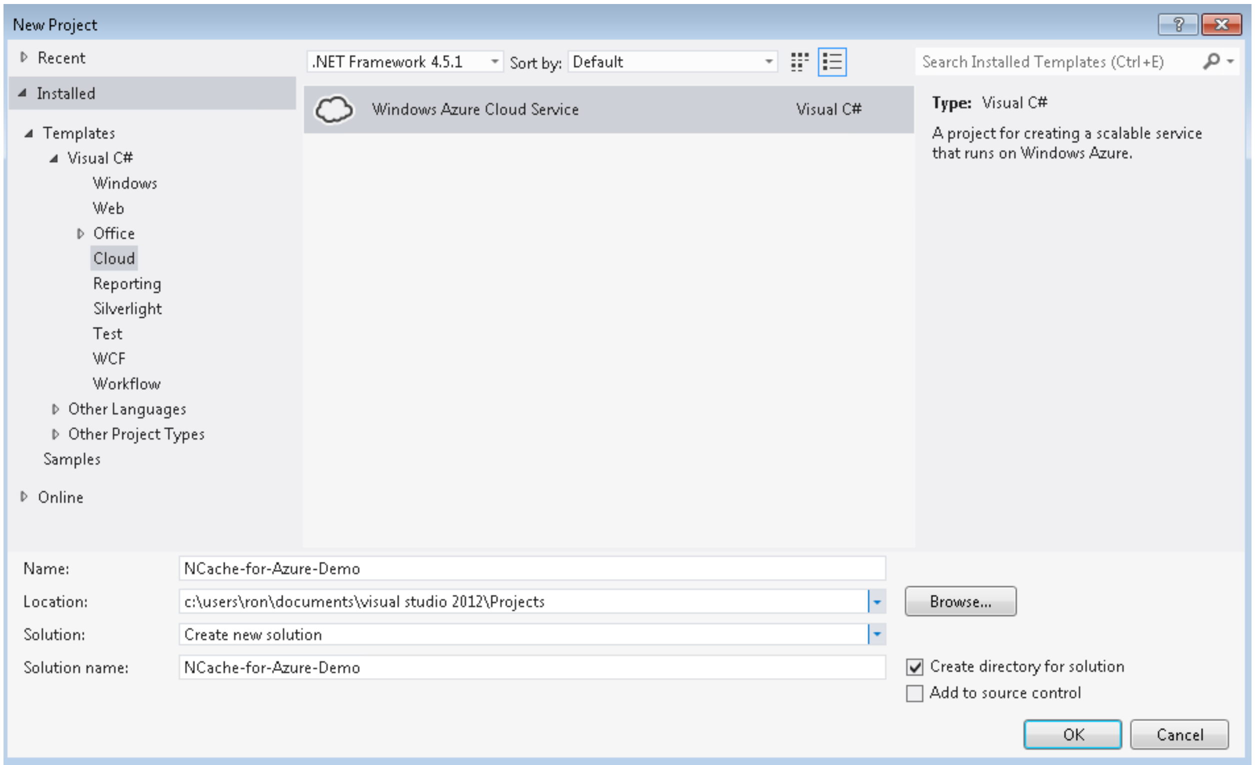
Microsoft Azure provides Infrastructure as a Service (IaaS) in the form of Azure Virtual Machines. When NCache for Azure Clients use this service of Azure and deploy their .NET application directly on VMs, it acts as a regular client deployment. You can just log onto Azure VMs and then install NCache for Azure remote client or deploy client libraries directly. You can then use NCache for Azure for applications that are going to be deployed on this Virtual Machine. There are no special configurations involved for Azure in this option.

3. NCache for Azure Clients as Web Roles or Worker Roles

You can choose to deploy your .NET application as a Web Role or Worker Role. NCache for Azure client libraries are deployed along with your application. The steps for using NCache for Azure in Web Role and Worker Role are the same. For demonstration purposes, here is a step by step guide on how to use NCache for Azure in Microsoft Azure project using an ASP.NET Web Role.

3.1. Create a Microsoft Azure Project

Create a new project and choose **Cloud** from installed templates and then name it.



Type: Visual C#

A project for creating a scalable service that runs on Windows Azure.

Name: NCache-for-Azure-Demo

Location: c:\users\ron\documents\visual studio 2012\Projects

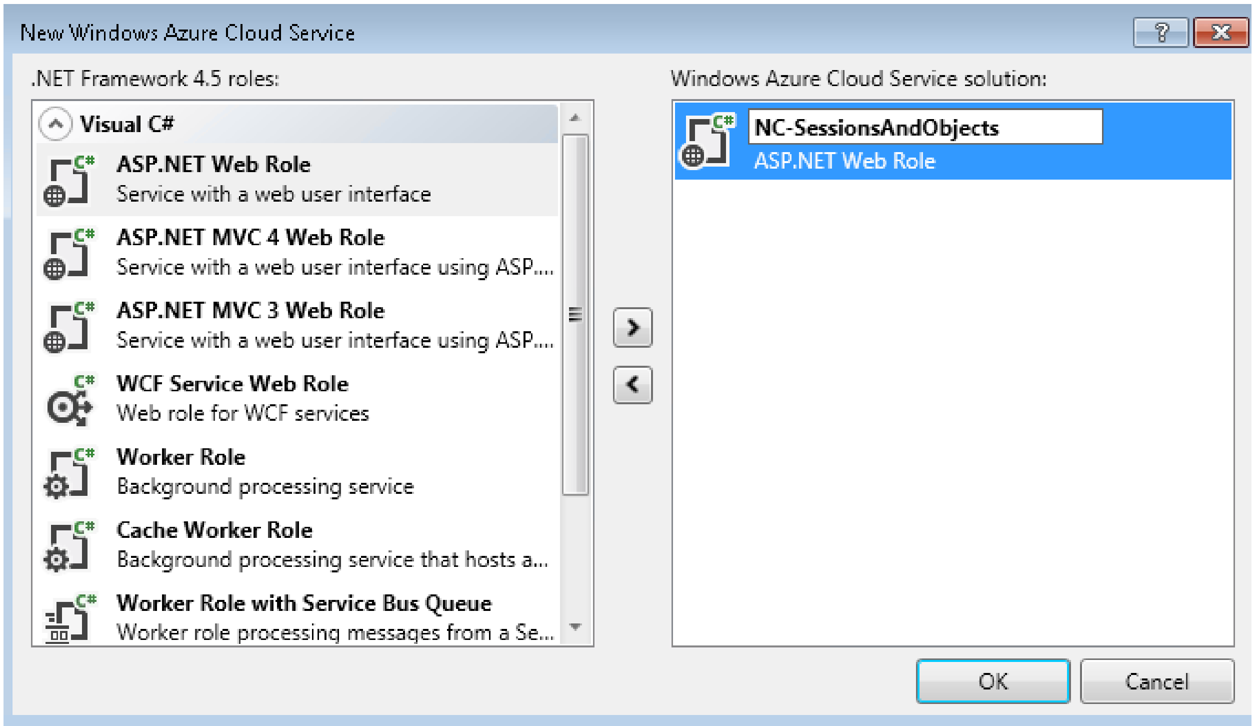
Solution: Create new solution

Solution name: NCache-for-Azure-Demo

☑ Create directory for solution

□ Add to source control

Choose **ASP.NET Web Role** on next screen and then choose **OK**.


New Windows Azure Cloud Service
? ✕
.NET Framework 4.5 roles:
Visual C#
ASP.NET Web Role
Service with a web user interface
ASP.NET MVC 4 Web Role
Service with a web user interface using ASP....
ASP.NET MVC 3 Web Role
Service with a web user interface using ASP....
WCF Service Web Role
Web role for WCF services
Worker Role
Background processing service
Cache Worker Role
Background processing service that hosts a...
Worker Role with Service Bus Queue
Worker role processing messages from a Se...
>
<
Windows Azure Cloud Service solution:
OK
Cancel

NC-SessionsAndObjects
ASP.NET Web Role

This will create a Microsoft Azure Cloud project with an ASP.NET Web Role.

3.2. Configure Network Settings of Web Role

If you opt for single site deployment where cache servers and clients are on same Azure Virtual Network then add Web Role network settings within the **ServiceConfiguration.Cloud.cscfg** file.

These settings will dictate that Web Role is also deployed on the same Azure Virtual Network and subnet where NCache for Azure servers are residing.

```
<NetworkConfiguration>
  <VirtualNetworkSite name="NCacheVirtualNetwork"/>
    <AddressAssignments>
      <InstanceAddress roleName="NC-SessionAndObjects">
        <Subnets>
           <Subnet name="NCacheSubnet" />
        </Subnets>
      </InstanceAddress>
    </AddressAssignments>
  </VirtualNetworkSite>
</NetworkConfiguration>
```

3.3. Equip Azure Web Role with NCache for Azure Assembly and Configuration Files

You now need to equip your Azure Web Role to be used with NCache for Azure. This requires NCache for Azure assembly reference, config files and code/configuration changes. You will also make necessary configurations and code changes to use NCache for Azure for session and object caching.

Here are two options for this:

- Using NCache for Azure NuGet Package
- Manually Configure NCache for with Web Role

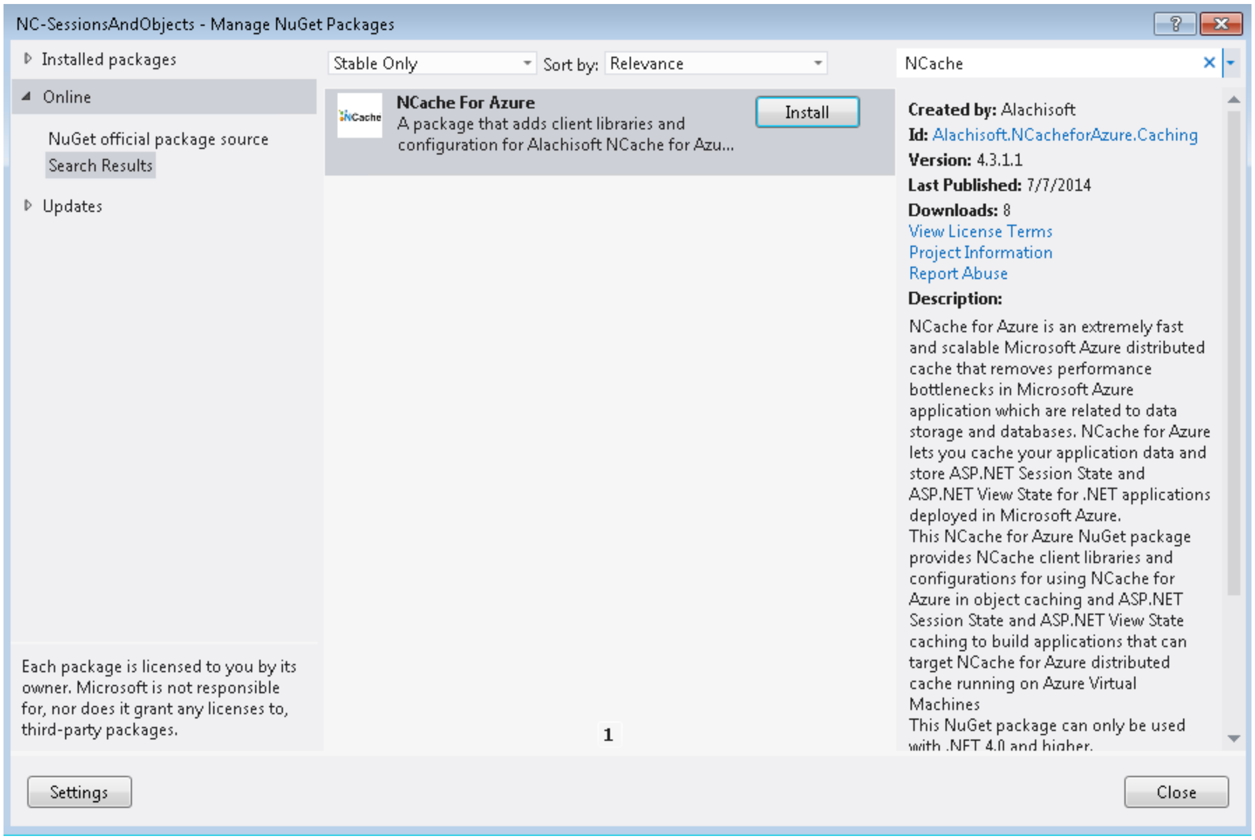
3.3.1. Using NCache for Azure NuGet Package

NCache for Azure provides a Microsoft Azure NuGet package that you can use to automatically configure your web role for NCache for Azure. This will automatically include references of all NCache for Azure and dependent assemblies along with NCache for Azure client configuration files. You can then change NCache for Azure configurations as per your requirements.

Please follow the steps given below:

- **Install NCache for Azure NuGet Package**

You can either download NCache for Azure NuGet project directly from our website or from NuGet gallery. You then need to add it in Visual Studio using NuGet Package manager and install it in the project.This will automatically include all required NCache for assemblies and configuration files in the Azure project.



Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.

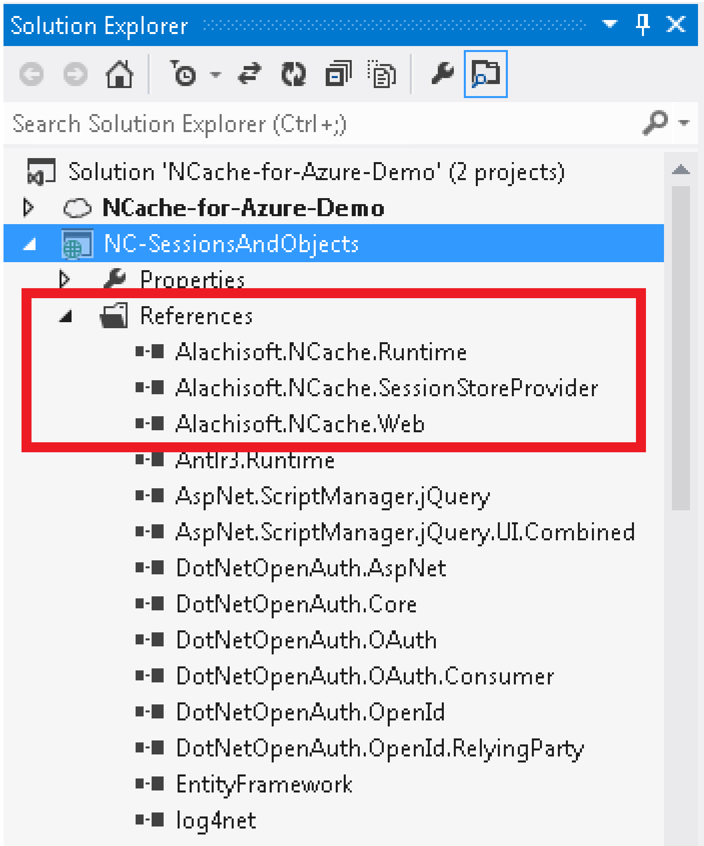
NCache For Azure
A package that adds client libraries and configuration for Alachisoft NCache for Azu...

Created by: Alachisoft
Id: Alachisoft.NCacheforAzure.Caching
Version: 4.3.1.1
Last Published: 7/7/2014
Downloads: 8
View License Terms
Project Information
Report Abuse
Description:
NCache for Azure is an extremely fast and scalable Microsoft Azure distributed cache that removes performance bottlenecks in Microsoft Azure application which are related to data storage and databases. NCache for Azure lets you cache your application data and store ASP.NET Session State and ASP.NET View State for .NET applications deployed in Microsoft Azure.
This NCache for Azure NuGet package provides NCache client libraries and configurations for using NCache for Azure in object caching and ASP.NET Session State and ASP.NET View State caching to build applications that can target NCache for Azure distributed cache running on Azure Virtual Machines
This NuGet package can only be used with .NET 4.0 and higher.

- **Verify NCache for Azure Assemblies**

Verify the following NCache for Azure assembly references in project references as soon as NCache for Azure NuGet package is installed within the project.

- **Alachisoft.NCache.Runtime**
- **Alachisoft.NCache.SessionStateProvider**
- **Alachisoft.NCache.Web**



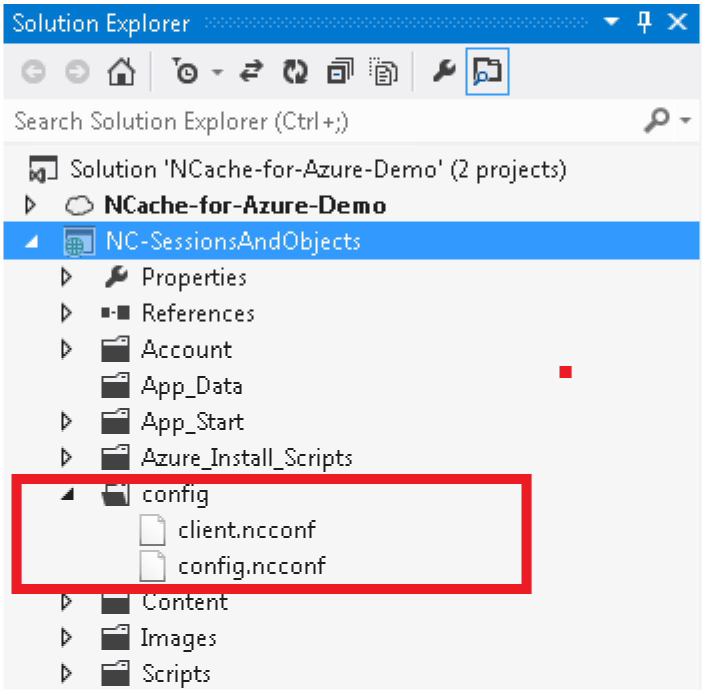
Solution 'NCache-for-Azure-Demo' (2 projects)
NCache-for-Azure-Demo
NC-SessionsAndObjects
Properties
References
Alachisoft.NCache.Runtime
Alachisoft.NCache.SessionStoreProvider
Alachisoft.NCache.Web
Antlr3.Runtime
AspNet.ScriptManager.jQuery
AspNet.ScriptManager.jQuery.UI.Combined
DotNetOpenAuth.AspNet
DotNetOpenAuth.Core
DotNetOpenAuth.OAuth
DotNetOpenAuth.OAuth.Consumer
DotNetOpenAuth.OpenId
DotNetOpenAuth.OpenId.RelyingParty
EntityFramework
log4net

Please also make sure that the project includes the following dependent assemblies as well.

- **Log4Net.dll**
- **Protobuf-net.dll**
- **Oracle.DataAccess.dll**

- **Verify NCache for Azure Configurations**

Verify that a config folder with NCache for Azure config files, **client.ncconf** and **config.ncconf**, is added in the project.



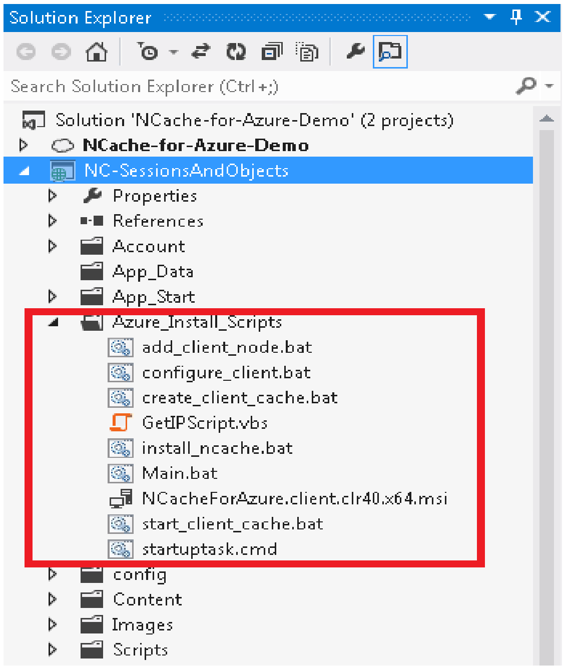
Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'NCache-for-Azure-Demo' (2 projects)
  NCache-for-Azure-Demo
  NC-SessionsAndObjects
    Properties
    References
    Account
    App_Data
    App_Start
    Azure_Install_Scripts
    config
      client.ncconf
      config.ncconf
    Content
    Images
    Scripts
```

- **NCache for Azure Setup and Startup Scripts**

NCache for Azure NuGet package includes NCache for Azure setup and startup scripts.



Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'NCache-for-Azure-Demo' (2 projects)
  NCache-for-Azure-Demo
  NC-SessionsAndObjects
    Properties
    References
    Account
    App_Data
    App_Start
    Azure_Install_Scripts
      add_client_node.bat
      configure_client.bat
      create_client_cache.bat
      GetIPScript.vbs
      install_ncache.bat
      Main.bat
      NCacheForAzure.client.clr40.x64.msi
      start_client_cache.bat
      startuptask.cmd
    config
    Content
    Images
    Scripts
```

The first script to run is **startuptask.cmd** which in turn calls various other scripts, performing the following major tasks:

- Install NCache for Azure setup
- Addclient node
- Create client cache if it is configured.
- Check if client cache is InProc or OutProc. If it is OutProc, then start it.

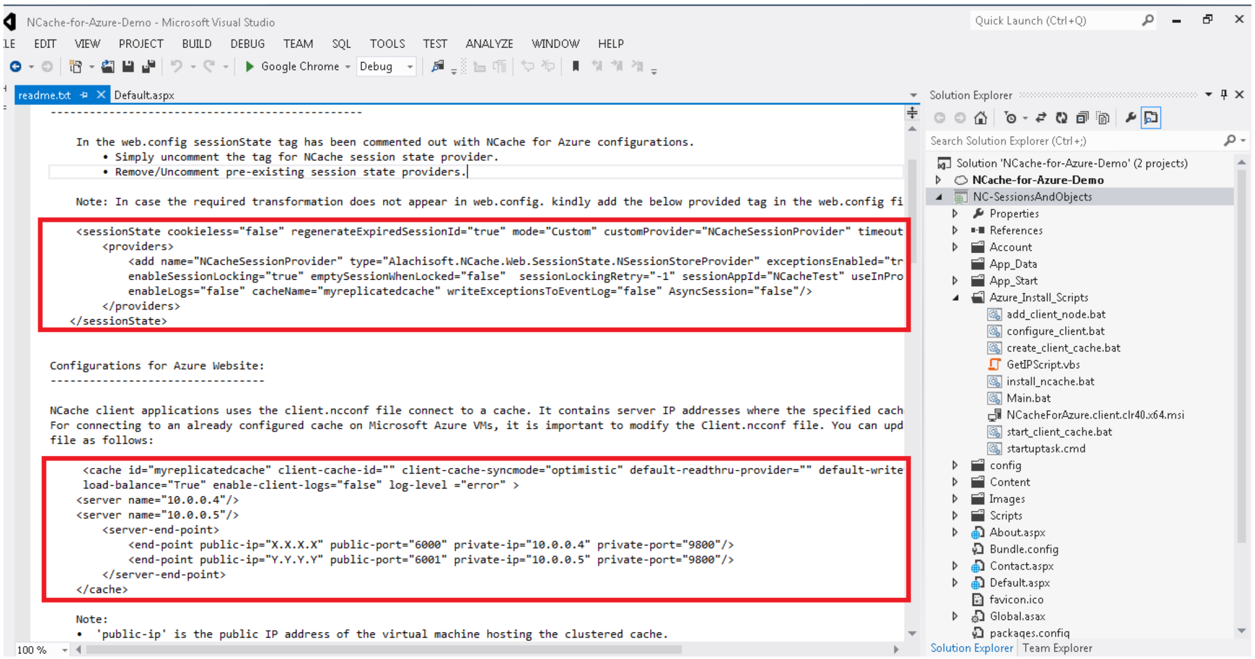
There are some startup environment variables used in these scripts which must be defined first. For that, open the **ServiceDefinition.csdef** file of your Azure Web Role and insert the following code in it:

```
    <Startup>

<TaskcommandLine="Azure_Install_Scripts\startuptask.cmd"executionContext="elevated"taskType="simple">
        <Environment>
          <!--Check whether Azure project is running in emulated environment or not.
If this role is running in emulated environment, then startup scripts will be avoided-->
          <Variablename="EMULATED">
            <RoleInstanceValuexpath="/RoleEnvironment/Deployment/@emulated" />
          </Variable>
          <!--Specify path for NCache Azure Client Setup for installation on client role-->

<Variablename="NCache.SetupPath"value="NCache4.3_Enterprise_DotNet_CLD_LWx64" />
          <!--Specify cache name to add current role as client node of specified cluster cache..-->
          <Variablename="NCache.CacheName"value="PartitionedCache" />
          <!--Specify cache server IP for communication with cache.
          For multiple servers, specify comma separated IPs as specified in example.
          For public port, specify ports with server as
"10.0.0.4:8270,10.0.0.5:8271"-->

<Variablename="NCache.CacheServers"value="23.97.66.31:8250,23.97.66.31:8251" />
          <!--Specify "true” if you want to create a client cache on this role otherwise "false"-->
          <Variablename="NCache.CreateClientCache"value="true" />
          <!--Specify client cache name if you want to configure client cache on this role-->
          <Variablename="NCache.ClientCacheName"value="myclientcache" />
          <!--Specify client cache size. (Compulsory option for configuring client cache)-->
          <Variablename="NCache.ClientCacheSize"value="250" />
          <!--Specify "true" if your client role is outside the virtual network of your cache.
          Your public endpoints will be used for communication with the server from current client role.
          If your client role is within the same virtual network as your cache, then specify "false"-->
          <Variablename="NCache.UseEndPoint"value="true" />
          <!--Specify startup task log path-->
          <Variablename="NCache.LogPath"value="%Temp%" />
          <!-- Specify if client cache is InProc or not-->
          <Variablename="NCache.IsInproc"value="false" />
        </Environment>
      </Task>
    </Startup>
```

A short description of each variable is provided in comments before its definition.

- **Refer to ReadMe file**

It also includes a **readme.txt** file that explains **web.config** changes for NCache for Azure session store provider and also NCache **client.ncconf** changes to connect to an already configured cache in your environment. This information can be used as a reference later on when you will change NCache for Azure configurations.



```
  ------------------------------------------------

    In the web.config sessionState tag has been commented out with NCache for Azure configurations.
        • Simply uncomment the tag for NCache session state provider.
        • Remove/Uncomment pre-existing session state providers.

    Note: In case the required transformation does not appear in web.config. kindly add the below provided tag in the web.config fi

    <sessionState cookieless="false" regenerateExpiredSessionId="true" mode="Custom" customProvider="NCacheSessionProvider" timeout
        <providers>
            <add name="NCacheSessionProvider" type="Alachisoft.NCache.Web.SessionState.NSessionStoreProvider" exceptionsEnabled="tr
            enableSessionLocking="true" emptySessionWhenLocked="false"  sessionLockingRetry="-1" sessionAppId="NCacheTest" useInPro
            enableLogs="false" cacheName="myreplicatedcache" writeExceptionsToEventLog="false" AsyncSession="false"/>
        </providers>
   </sessionState>


  Configurations for Azure Website:
  --------------------------------

  NCache client applications uses the client.ncconf file connect to a cache. It contains server IP addresses where the specified cach
  For connecting to an already configured cache on Microsoft Azure VMs, it is important to modify the Client.ncconf file. You can upd
  file as follows:

     <cache id="myreplicatedcache" client-cache-id="" client-cache-syncmode="optimistic" default-readthru-provider="" default-write
     load-balance="True" enable-client-logs="false" log-level ="error" >
    <server name="10.0.0.4"/>
    <server name="10.0.0.5"/>
        <server-end-point>
            <end-point public-ip="X.X.X.X" public-port="6000" private-ip="10.0.0.4" private-port="9800"/>
            <end-point public-ip="Y.Y.Y.Y" public-port="6001" private-ip="10.0.0.5" private-port="9800"/>
        </server-end-point>
    </cache>

    Note:
    •  'public-ip' is the public IP address of the virtual machine hosting the clustered cache.
```

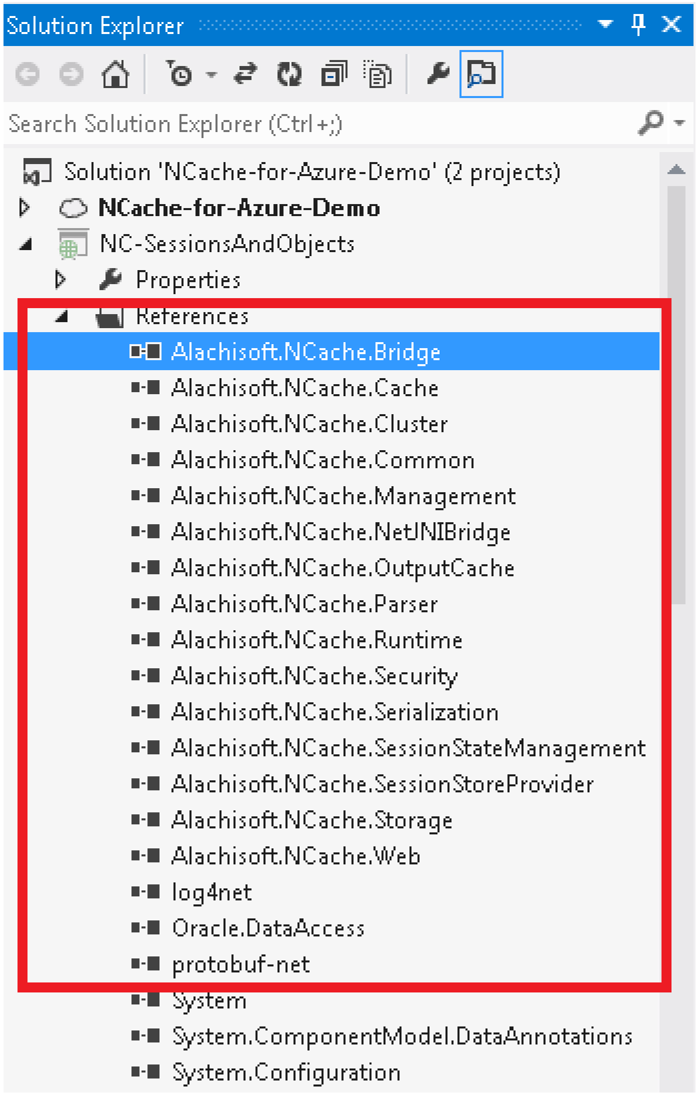
3.3.2. Manually Configure Web Role for NCache for Azure

If you do not want to use NCache for Azure NuGet package then you need to manually add all NCache for Azure and its dependent assemblies along with required NCache for Azure configuration files.

Here is a step by step detail on this:

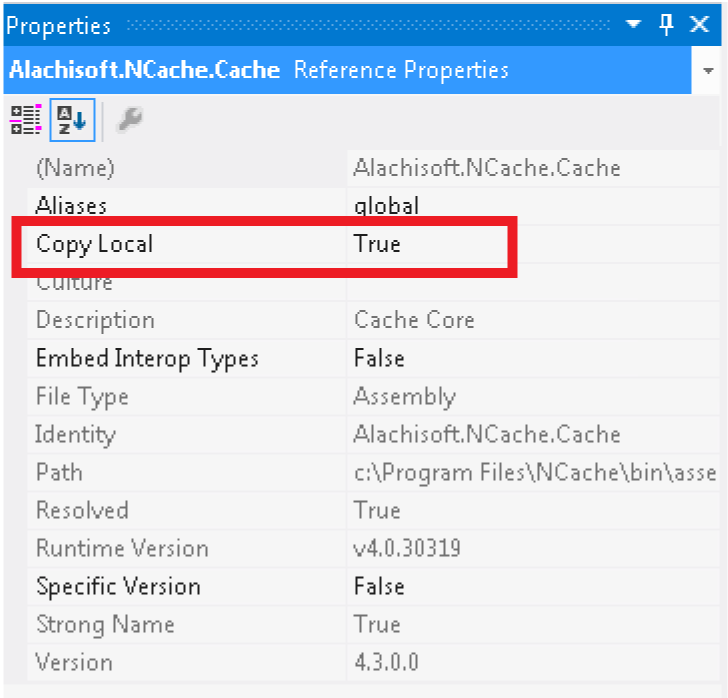
- **Add NCache for Remote Client Assembly References**

Add all NCache for client assembly references within your Microsoft Azure project. These assemblies can be downloaded from our website separately.



Solution 'NCache-for-Azure-Demo' (2 projects)
NCache-for-Azure-Demo
NC-SessionsAndObjects
Properties
References
Alachisoft.NCache.Bridge
Alachisoft.NCache.Cache
Alachisoft.NCache.Cluster
Alachisoft.NCache.Common
Alachisoft.NCache.Management
Alachisoft.NCache.NetJNIBridge
Alachisoft.NCache.OutputCache
Alachisoft.NCache.Parser
Alachisoft.NCache.Runtime
Alachisoft.NCache.Security
Alachisoft.NCache.Serialization
Alachisoft.NCache.SessionStateManagement
Alachisoft.NCache.SessionStoreProvider
Alachisoft.NCache.Storage
Alachisoft.NCache.Web
log4net
Oracle.DataAccess
protobuf-net
System
System.ComponentModel.DataAnnotations
System.Configuration

Now right click on the assemblies, select **properties** and set **Copy Local** to **True** for all NCache for Azure Assemblies.

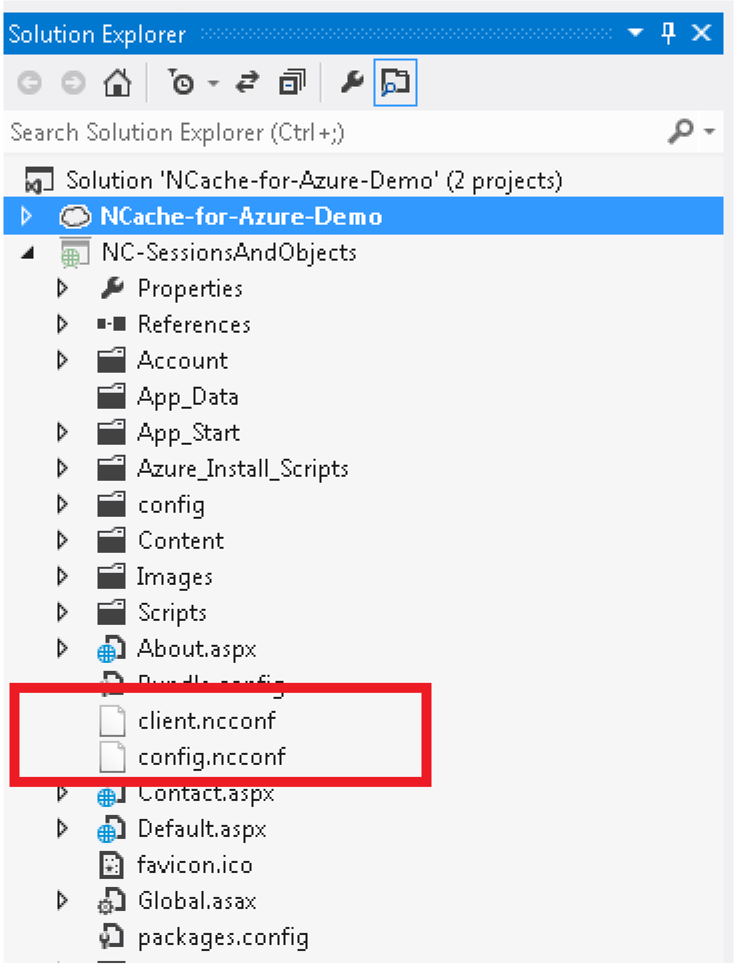


Alachisoft.NCache.Cache Reference Properties

(Name)	Alachisoft.NCache.Cache
Aliases	global
Copy Local	True
Culture	
Description	Cache Core
Embed Interop Types	False
File Type	Assembly
Identity	Alachisoft.NCache.Cache
Path	c:\Program Files\NCache\bin\asse
Resolved	True
Runtime Version	v4.0.30319
Specific Version	False
Strong Name	True
Version	4.3.0.0

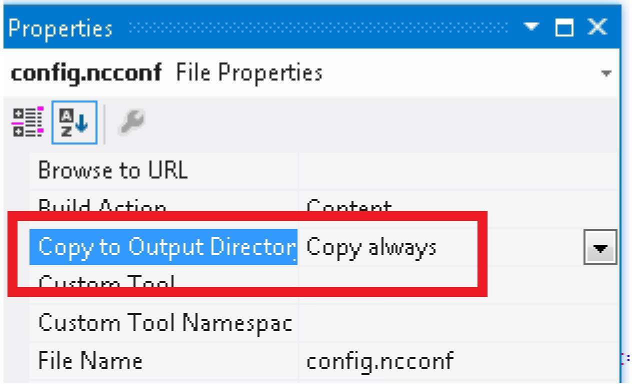
- **Include client.ncconf file in the Web Role**

Add NCache for Azure configuration files**, config.ncconf** and **client.ncconf**, manually within your Microsoft Azure project. These come installed with NCache for Azure installation and can be copied from any box.



```
Solution 'NCache-for-Azure-Demo' (2 projects)
▷ NCache-for-Azure-Demo
◢ NC-SessionsAndObjects
    ▷ Properties
    ▷ References
    ▷ Account
      App_Data
    ▷ App_Start
    ▷ Azure_Install_Scripts
    ▷ config
    ▷ Content
    ▷ Images
    ▷ Scripts
    ▷ About.aspx
      Bundle.config
      client.ncconf
      config.ncconf
    ▷ Contact.aspx
    ▷ Default.aspx
      favicon.ico
    ▷ Global.asax
      packages.config
```

Right-click on the **client.ncconf** and **config.ncconf** file and select **Properties**. Set the **Copy to Output Directory** to **Copy Always.**



config.ncconf File Properties

Browse to URL	
Build Action	Content
Copy to Output Director	Copy always
Custom Tool	
Custom Tool Namespac	
File Name	config.ncconf

3.4. Modify NCache for Azure client.ncconf File to Connect to a Cache

The file **client.ncconf** is used to connectto any clustered cache. It contains server IP addresses where clustered cache resides. You need to modify client.ncconf file so that it canconnect to an already configured cache on Azure VMs.

Here is how you should update this file:

3.4.1. For Single-Site Deployment

If the Web Role is going to be deployed in the same Azure Virtual Network as the Virtual Machines, then there are no special modifications involved.

Here is a sample cache tag for cache named Democache and is configured on Azure VMs with IP addresses 10.0.0.4 and 10.0.0.5.

```
<cache id="Democache"client-cache-id="Democlientcache"client-cache-
syncmode="optimistic"default-readthru-provider="ReadthruProvider"default-writethru-
provider="WritethruProvider"load-balance="True">
  <servername="10.0.0.4"/>
  <servername="10.0.0.5"/>
</cache>
```

Web Role should be able to connect to the cache without any explicit port forwarding.

3.4.2. For Multi-Site Deployment

If Web Role is going to be deployed outside the Azure Virtual Network of the cache servers, then you need to provide an additional server IP mapping tag in **client.ncconf** file. This information will be used to connect to our cache from outside its Azure Virtual Network. Here is a sample cache tag to connect to our cache from outside its Azure Virtual Network:

```
<cache id="AzureCache"client-cache-id="Azureclientcache"client-cache-
syncmode="optimistic"default-readthru-provider="ReadthruProvider"default-writethru-
provider="WritethruProvider"load-balance="True">
  <servername="10.0.0.4"/>
  <servername="10.0.0.5"/>
    <server-ip-mapping>
      <mapping public-ip="207.46.129.19"public-port="9800"private-
      ip="10.0.0.4"private-port="9800"/>
      <mapping public-ip="207.46.129.19"public-port="9801"private-
      ip="10.0.0.5"private-port="9800"/>
    </server-ip-mapping>
</cache>
```

Following are details on the specified public and private IP addresses and ports:

- **Public IP Address:** Public IP address of the Virtual Machine is where a clustered cache is configured.
- **Public Port:** Public Port of the Virtual Machine starts with 9800 and is incremented by one for each cache server. This is the public end point defined for our respective Cache VM.
- **Private IP:** Itis the internal or actual IP Address of each Virtual Machine.
- **Private Port.** Itisthat port of NCache for Azure which is 9800 and is the same for all NCache for Azure VMs. It is defined as the private endpoint for our NCache for Azure VM.

3.5. Use NCache for Azure for Objects and Sessions Caching

3.5.1. Use NCache for Azure for Object Caching

If you need to use NCache for Azure for object caching where you'll be making NCache API calls, then you need to do the following in your Visual Studio Azure Web Role project C# or VB.NET source code.

- **Add NCache for Azure Namespaces**

It is required to include following namespaces in the source code to use NCache for Azure client API:

```
using Alachisoft.NCache.Web.Caching;

using Alachisoft.NCache.Runtime;
```

- **Include NCache for Azure API to Test the Cache**

Add following code to quickly test the cache:

```
  _cache = NCache.InitializeCache("DemoCache");

//Add items in the cache
for (int i=0; i<100;i++)
            {
                _cache.Insert("Key" + i.ToString(), "DummyObject" + i.ToString());
            }
```

3.5.2. Use NCache for Azure for ASP.NET Session Caching

If you're using NCache for Azure for ASP.NET Sessions, you need to modify your **web.config**. You need to add NCache for Azure Session State Provider (SSP) configurations in your **web.config** as followed:

```
<compilation defaultLanguage ="C#" debug="true" >
  <assemblies>
       <add assembly="Alachisoft.NCache.SessionStoreProvider,
       Version=4.1.0.0,Culture=neutral,
       PublicKeyToken=CFF5926ED6A53769"/>
  </assemblies>
</compilation>
```

Note: Make sure that **Version** above reflects the correct version of NCache for Azure.

```
<sessionState cookieless = "false" regenerateExpiredSessionId = "true" mode = "Custom
" customProvider = "NCacheSessionProvider" timeout= "20">
  <providers>
       <add name = "NCacheSessionProvider" type = "Alachisoft.NCache.Web.SessionState
       .NSessionStoreProvider" sessionAppId ="NCacheTest" cacheName = "myCache" write
       ExceptionsToEventLog = "false"/>
  </providers>
</sessionState>
```

3.6. Test Web Role Locally or Deploy in Azure Environment

Visual Studio lets you emulate cloud environment and it can be tested locally or it can be deployed in Azure environment and tested there. You can test the Web Role once you have completedthe above steps and verify that it works.