



Sitecore Azure 1.1.0

Getting Started with

Sitecore Azure

An Overview for Sitecore Administrators

Table of Contents

Chapter 1	Getting Started with Sitecore Azure	3
1.1	Overview	4
1.1.1	Prerequisites for Sitecore Azure Installation.....	4
1.1.2	Using a Microsoft Azure Subscription vs. a Sitecore-Managed Azure Subscription	5
1.1.3	How to Install Sitecore Azure	5
1.2	Using Sitecore Azure.....	8
1.2.1	How to Create a Sitecore Azure Content Delivery Environment.....	8
1.2.2	How to Publish to a Sitecore Azure Content Delivery Environment.....	9
1.2.3	How to Deploy Files to an Environment	9
1.2.4	How to Change the Number of Instances in an Environment	10
1.2.5	How to Suspend or Start the Instances in an Environment	10
1.2.6	How to Remove Instances from an Environment	10
1.2.7	How to Create a Sitecore Azure Content Delivery Environment for Multiple Sites	10
1.3	Answers to Common Questions.....	12
1.3.1	How Do Sitecore Azure Content Delivery Environments Clear Caches?	12
1.3.2	Where does Sitecore Azure Store Configuration?	12
1.3.3	How Does Sitecore Azure Assemble a Solution to Deploy?.....	13
1.3.4	How Do I Use My Own Domain Name?	13
1.3.5	How to Encrypt Your Package.....	13
1.3.6	How to Add AppFabric Caching	14
1.3.7	How to Enable SSL in a Sitecore Azure Project	14
1.3.8	How to Enable Remote Desktop Access to Webrole Instances	15
1.3.9	How to Add or Replace Management Certificates	15
1.3.10	Environment File Deployment Naming Convention	17
1.3.11	How to Configure Firewall Rules	18
1.3.12	How to Create a Database Account for Sitecore Azure on SQL Server	18
1.3.13	How to Create a Database Account for Sitecore Azure on SQL Azure Server.....	19
1.4	Known Issues, Considerations, and Troubleshooting.....	20
1.4.1	ASP.NET Considerations on Windows Azure	20
1.4.2	SQL Azure Considerations.....	20
1.4.3	Sitecore Azure Considerations	20
1.4.4	Troubleshooting Sitecore Azure	21
1.4.5	Creating a Self-Signed Certificate	21
1.4.6	How to Replace a Certificate.....	22

Chapter 1

Getting Started with Sitecore Azure

This document introduces Sitecore Azure. Sitecore administrators should read this document before installing or using Sitecore Azure.

Beginning with an overview of the underlying Windows Azure and Sitecore Azure concepts and terminology, this document describes procedures for accomplishing specific tasks with Sitecore Azure, attempts to answer common questions about Sitecore Azure, and then provides information about known issues and troubleshooting techniques.

This document contains the following sections:

- Overview
- Using Sitecore Azure
- Answers to Common Questions
- Known Issues, Considerations, and Troubleshooting

1.1 Overview

Sitecore Azure provides a set of software and services to simplify the deployment of your web solutions to the Windows Azure cloud. You can use Sitecore Azure with the Sitecore CMS to host Sitecore solutions in the Windows Azure cloud.

Windows Azure is an edition of the Microsoft Windows operating system that runs in an Internet cloud. The Windows Azure fabric consists of hardware and software to manage that hardware. The fabric supports web roles, which provide IIS, and worker roles, which do not. Roles correspond to virtual machines in the cloud. Each web role hosts a single IIS Web site (usually a single Sitecore instance).

The Windows Azure fabric coordinates some number of roles to implement each task, such as hosting a website. You can acquire Windows Azure services from Microsoft and manage deployment of your own Web Applications, or you can use Sitecore Azure to manage your cloud content delivery infrastructure.

SQL Azure is an implementation of Microsoft SQL Server that runs on Windows Azure. SQL Azure presents some limitations in comparison with traditional SQL Server.

With Sitecore Azure, you can deploy from one or more internal content management servers to any number of Sitecore Azure content delivery instances hosted at the various Windows Azure locations. Sitecore Azure provides automation and a browser-based user interface to manage your Windows Azure deployment.

Sitecore Azure provides the following benefits:

- Sitecore Azure provides for extreme scalability.
- Sitecore Azure significantly reduces the costs of providing geographically distributed load balancing and redundancy between geographies for performance, disaster preparedness, and other purposes.
- Sitecore Azure reduces the cost and time required to add and remove instances from server farms in order to meet expected load.
- Sitecore Azure automates much of the process of migrating a website to Windows Azure.

1.1.1 Prerequisites for Sitecore Azure Installation

You must have the following requirements in place before installing Sitecore Azure.

- Sitecore Azure requires a 64-bit edition of Microsoft Windows.
- You must have a Windows Live ID.
- Sitecore Azure requires Sitecore CMS version 6.3 or higher.
- For SQL Server installations, Sitecore Azure requires SQL Server 2008, Service Pack 1.
- Sitecore Azure requires that you install the Windows Azure SDK version 1.6 in the internal content management environment.

For more information and to download the Windows Azure SDK version 1.6, see <http://www.microsoft.com/windowsazure/sdk/>.

- You must configure your web server's firewall rules to allow Sitecore Azure traffic.

For more information see the section *How to Configure Firewall Rules*.

- You must request a Sitecore Azure environment file from Sitecore. See the section *How to Install Sitecore Azure*

1.1.2 Using a Microsoft Azure Subscription vs. a Sitecore-Managed Azure Subscription

When you purchase a Sitecore Azure subscription you receive the benefits of Sitecore's automated features for managing your Azure installations. These benefits include automated set-up of subscriptions and databases. You will also receive licensing benefits such as bursting and prioritized support. Your local Sitecore office can provide more details about the licensing benefits.

If you already have an existing Microsoft Azure subscription that you want to use for your Sitecore installation, you can provide that information to Sitecore and we will assist you in your installation. See the section **Error! Reference source not found.**

If you are using a Microsoft Azure subscription, you will still get many of the benefits of using Sitecore's Azure tools, but you will have to install and manage your own certificates.

1.1.3 How to Install Sitecore Azure

To install Sitecore Azure, you need a Sitecore CMS license and a Sitecore installation. To use Sitecore Azure, you will need a Sitecore Azure environment file. If you have not already done so, contact Sitecore in your region to obtain software and licensing. To contact Sitecore, see <http://www.sitecore.net/contact.aspx>.

Sitecore environment files are configuration files that contain information about your Windows Azure subscription. You can have more than one environment file so that you can support different kinds of environments, for example, you may have separate development, testing and production environments. Sitecore Azure allows you to manage all of these environments from within the Sitecore Azure application.

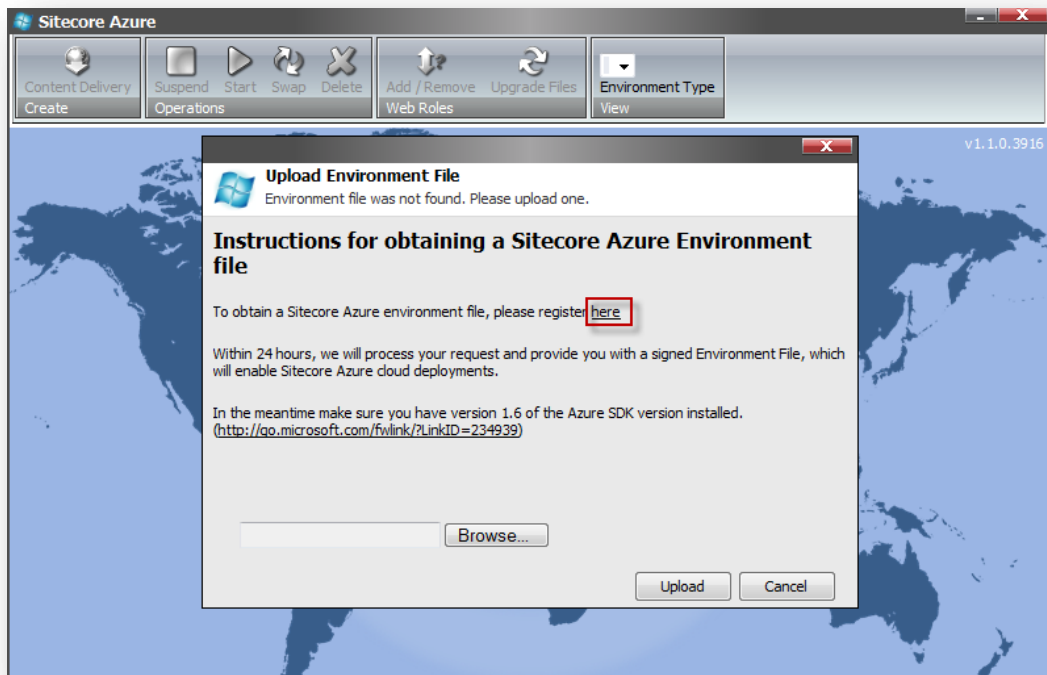
Download the Sitecore Azure installation package from <http://sdn.sitecore.net/Products/Sitecore%20Azure/Download%20Sitecore%20Azure.aspx>.

To install Sitecore Azure:

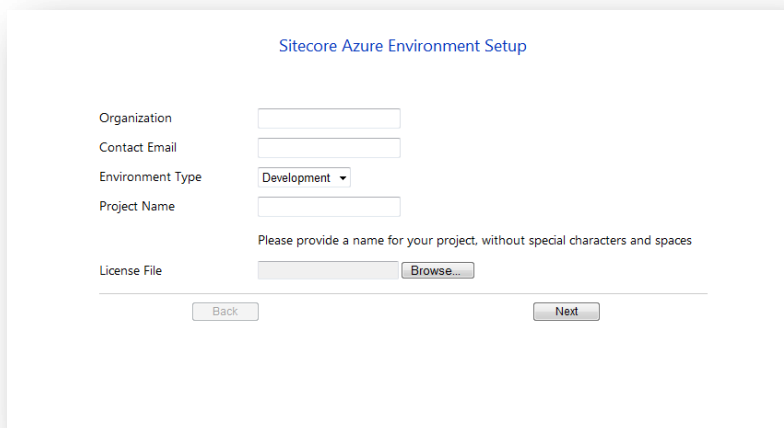
1. Log in to the desktop of your internal content management environment as an administrator.
2. In the Sitecore desktop, click **Sitecore, Development Tools, Installation Wizard**, and then use the **Installation Wizard** to install the *Sitecore Azure* package, overwriting any existing components when prompted.
3. In the Sitecore desktop, click **Sitecore**, and then click **Sitecore Azure. Sitecore Azure** appears prompting you to upload the Sitecore Azure Environment file that you have received from Sitecore.

If this is a first time installation and you have not already obtained a Sitecore environment file; click on the link shown in the dialog box to submit your information to Sitecore and you will

receive an environment file within 24 hours.

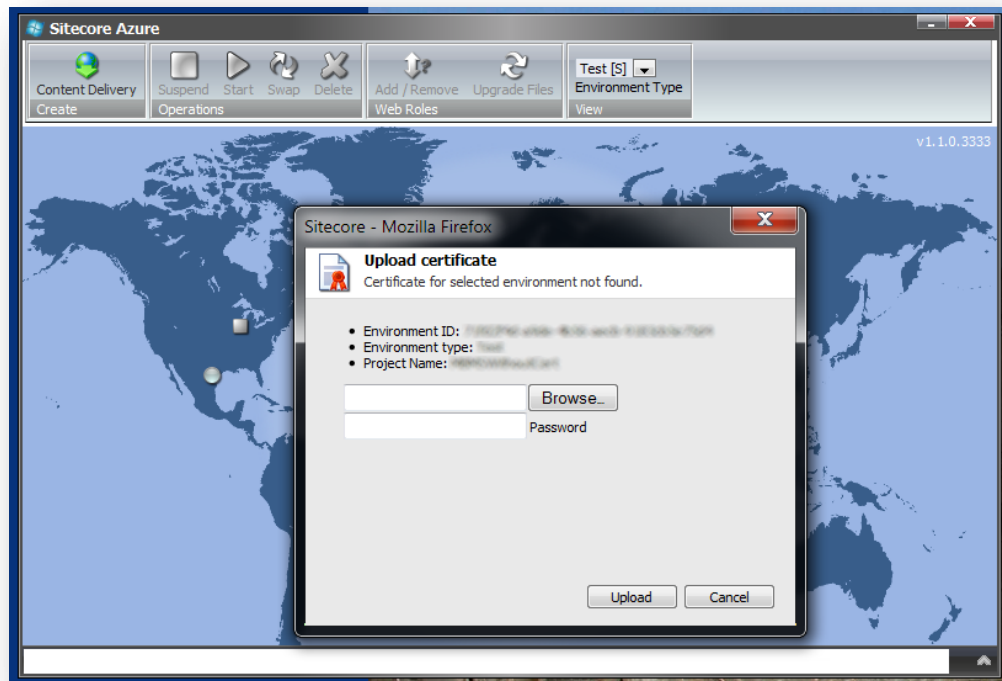


This link to Sitecore’s web site will present a page that asks you for basic contact information, a project name, and to upload your Sitecore license file.



- Once you have obtained your environment file, you can continue with the Sitecore Azure Set-up dialog box, click **Browse ...** and locate the *Environment* file you obtained from Sitecore. Once you have selected the file, click **Upload** to continue. Your environment file name will be in the format of `ProjectName<Test>_Environment.xml`.

5. In the Sitecore Azure application, select a location and click **Content Delivery**. You will be prompted to upload a certificate.



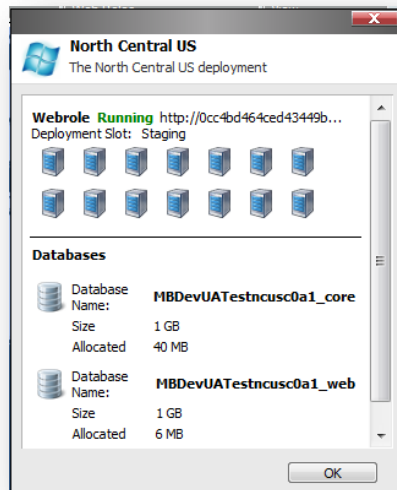
6. Select certificate .pfx file and provide the password then click **Upload**.

If you do not already have a certificate from a certificate authority installed on your web server you can create a self-generated certificate for development and testing purposes. See the section *Creating a Self-Signed Certificate*.

1.2 Using Sitecore Azure

Sitecore Azure provides a map overlaid with icons that represent available Windows Azure hosting locations. Each icon and its color indicate the status of the project in that environment:

- **Gray Circle:** Unused. You can create a content delivery environment.
- **Green Circle:** Started. You can access this content delivery environment over the internet.
- **Red Circle:** Suspended. You can start this content delivery environment.
- **Hourglass:** Busy. Sitecore Azure is attempting to start, suspend, delete, or otherwise administer the location, or the current operation for the location is queued.
- When you click an icon to select a location, Sitecore Azure displays a status dialog box for that location. Ribbon commands operate on the selected location. When you click on a location to select it, its icon will change from a circle to a square. You can open multiple Sitecore Azure windows in the Sitecore desktop to access the status of tasks in multiple locations.



Warning

Certain operations in the Sitecore Azure user interface, such as, adding or removing an SQL Azure publishing target database, updating configuration, and so on, cause ASP.NET to restart. These operations can therefore interfere with other users and processes on the system. Deployment operations will also cause an ASP.NET restart during the final stage. If you attempt to run multiple deployments concurrently, Sitecore will wait until all deployments are finished to trigger the ASP.NET restart and for the changes in all deployments to be available.

Note

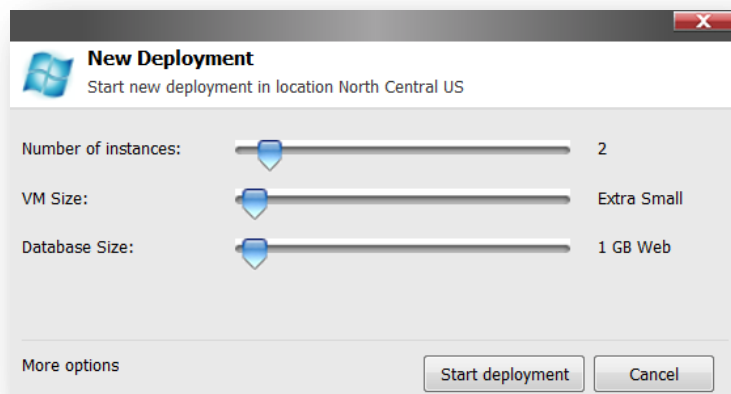
The Sitecore Azure user interface can seem unresponsive because of long-running remote service calls.

1.2.1 How to Create a Sitecore Azure Content Delivery Environment

For information about how Sitecore Azure assembles a solution to deploy, see the section *How Does Sitecore Azure Assemble a Solution to Deploy?*

To create a Sitecore Azure content delivery environment:

1. To avoid the need to publish after deployment, before creating a Sitecore Azure content delivery environment, publish to the default publishing target named Internet (the Web database).
2. To minimize deployment time, remove unnecessary files from the document root of the content management environment.
3. In **Sitecore Azure**, click the gray circle that represents a location to which you would like to deploy your solution.
4. In the **Create** group, click **Content Delivery**.



5. In the **New Deployment** dialog box, make selections for Number of Instances (at least 2 is recommended), VM Size (Extra Small is fine for Evaluation and Test deployments) and Database Size (1GB is fine for Evaluation and Test deployments). Click Start Deployment.

The Hourglass icon will display. You can expand the status pane at the bottom of the Sitecore Azure window to see errors, warnings and informational messages while the deployment is running.

6. When deployment completes, click the URL at the top of the status dialog box for the location to open the deployed solution in a new browser window.

By default, Sitecore Azure allocates a single web role in each location. For information about changing the number of web roles in a location, see the section *How to Change the Number of Instances in an Environment*.

Note

The length of the deployment process can vary greatly between environments depending on the size of the solution, network conditions, and other factors.

1.2.2 How to Publish to a Sitecore Azure Content Delivery Environment

To publish to a Sitecore Azure content delivery environment:

- Publish to the publishing target database associated with the Windows Azure location.

1.2.3 How to Deploy Files to an Environment

To deploy files to a Sitecore Azure content delivery environment:

1. In **Sitecore Azure**, click the icon that represents the location. The status dialog box for that location appears.
2. In **Sitecore Azure**, in the **Web Roles** group, click **Upgrade Files**.

1.2.4 How to Change the Number of Instances in an Environment

To change the number of Sitecore instances in a Sitecore Azure content delivery environment:

1. In **Sitecore Azure**, click the icon that represents the location. The status dialog box for that location appears.
2. In **Sitecore Azure**, in the **Web Roles** group, click **Add/Remove**. A text field appears containing the number of instances that currently exist at this location.
3. Enter the number of instances to configure at this location, and then click **Change**.

Note

Microsoft recommends that you create at least two web role instances in each location. When there are two or more web roles instances, there is no down time when you upgrade the files.

1.2.5 How to Suspend or Start the Instances in an Environment

To suspend the Sitecore content delivery instances (Windows Azure web roles) in an environment:

1. In **Sitecore Azure**, click the icon that represents the location. A dialog box appears indicating the status of the current task for that location.
2. In **Sitecore Azure**, in the **Operations** group, click **Suspend**.

To start the instances in an environment:

1. In **Sitecore Azure**, click the icon that represents the location. A dialog box appears indicating the status of the current task for that location.
2. In **Sitecore Azure**, in the **Operations** group, click **Start**.

1.2.6 How to Remove Instances from an Environment

To remove all instances from an environment:

1. In **Sitecore Azure**, click the icon that represents the location.
2. In **Sitecore Azure**, in the **Operations** group, click **Suspend**.
3. When the **Suspended** status appears, in the **Operations** group, click **Delete**.

1.2.7 How to Create a Sitecore Azure Content Delivery Environment for Multiple Sites

For information about how create multiple sites in Sitecore CMS, see the article *Configuring Multiple Sites* (<http://sdn.sitecore.net/Articles/Administration/Configuring%20Multiple%20Sites.aspx>) on SDN.

To create a Sitecore Azure content delivery environment for multiple sites:

1. Configure a Web Role for Multiple Web Sites in the `ServiceDefinition.csdef` file. The value of the `ServiceDefinition.csdef` file is stored in the deployment item, in the `ServiceDefinition` field.

Under `/sitecore/system/Settings/Azure/Environments` there are one or more `Environments` items which represent installed environments. Under this item are deployment items which represent settings for deployment. You can access this through the Content Editor or by using the **More Options** button in the **New Deployment** dialog box.

Note

Please look at the following article about how to Configure a Web Role for Multiple Web Sites in Windows Azure via `ServiceDefinition.csdef` file: <http://msdn.microsoft.com/en-us/library/windowsazure/gg433110.aspx>

2. In the ServiceDefinition field, under the Sites element for the WebRole, copy the following XML code after the existing Site element:

```
<Site name="web-site-name" physicalDirectory=".\\SitecoreWebSite">
  <Bindings>
    <Binding name="HttpIn" endpointName="HttpIn" hostHeader="custom-domain-name" />
  </Bindings>
</Site>
```

3. Fill out the name attribute of the Site element with a unique value.
4. Fill out the hostHeader attributes of the Binding element with a host name for your new web site.
5. Save changes in the AzureDeployment item.

Note

The physicalDirectory attribute must be the same for all Site elements.

The hostHeader attribute must be filled out for all Binding elements.

The hostHeader attribute value should be the same as it is in the hostName attribute for a logical site in the web.config file under the Sites section.

A deployment is now configured to create a multiple sites solution in Sitecore Azure Content Delivery Environment.

In Sitecore Azure, select your deployment on the map and click the Upgrade Files button in the ribbon.

If you deploy a new solution, come back to the Sitecore Azure and click Start deployment button.

To test Sitecore Azure content delivery environment for multiple sites on local machine:

1. In **Sitecore Azure**, double click the running deployment on the map. The **Status** dialog box for a location appears.
2. In the **Status** dialog box, copy the value of your Windows Azure host name from the **Webrole** section.
3. In the Windows host file, map your hostname (hostname and hostHeader attributes value) to your Windows Azure host name.

Note

Please look at the following article about Overview of Managing Deployments in Windows Azure:
<http://msdn.microsoft.com/en-us/library/windowsazure/hh386336.aspx>

1.3 Answers to Common Questions

This section provides answers to some common questions.

1.3.1 How Do Sitecore Azure Content Delivery Environments Clear Caches?

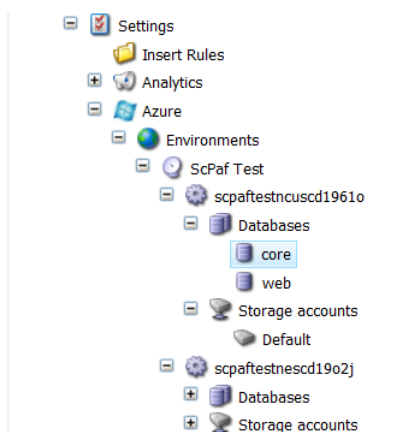
After completing a publishing operation, the Sitecore Azure content management environment writes a timestamp to the publishing target database. Sitecore Azure content delivery environments check the timestamp on each request, and clear caches accordingly.

1.3.2 Where does Sitecore Azure Store Configuration?

Sitecore Azure stores configuration information in a number of locations. Sitecore Azure stores basic configuration details in the `/App_Config/Include/Sitecore.Azure.config` file.

Sitecore manages configuration information using the `/Sitecore/System/Settings/Azure` item and its descendants.

Under `/sitecore/system/Settings/Azure/Environments` there are one or more `Environments` items which represent installed environments. Under this item are deployment items which represent settings for deployment. You can access this through Content Editor or by using the More Options button in the New Deployment dialog box.



There are several fields which contain configuration information s for deployment, and can be changed manually for customization.

- Under each deployment item there is a Databases item which contains database projects for deployment to SQL Azure. By default there are “web” and “core” database. If you need to deploy another database to SQL Azure you can create another database item.
- Under each deployment item there is Storage accounts item which contains storage projects for deployment to Azure. There is “Default” storage, which is used for uploading a package before deploying. If you need custom storage for a diagnostic configuration you can add one with the predefined name “Diagnostics”. This storage will be created if it does not already exist and is used for diagnostics set up.
- During the “Preparing Environment” operation each SQL Azure database server and storage will be created if it does not already exist, and will be used during the deployment process.

1.3.3 How Does Sitecore Azure Assemble a Solution to Deploy?

To assemble the document root of a solution to deploy, Sitecore Azure performs the following tasks, not necessarily in this order.

- Sitecore Azure copies the document root subdirectory of the content management environment to the `Data\AzurePackages\ (xxx) Deployment\SitecoreWebSite` subdirectory of the directory specified by the `dataFolder` variable in the `web.config` file.
- Sitecore Azure copies the contents of the `Deployment item\OverrideSourcesFolder` subdirectory in the document root of the CM environment to the `Data\AzurePackages\ (xxx) Deployment\SitecoreWebSite` subdirectory.
- Sitecore Azure removes some unnecessary files and subdirectories from the `SitecoreWebSite` subdirectory.
- Sitecore Azure prepares `web.config` by including files from the include folder as in a normal installation, does an XSLT transformation based on the XSLT patch file which is stored in the `Deployment item\webConfigPatch` field. In order to decrease size of `web.config` by the end of preparation, the `web.config` is divided into several include files which are stored in the `App_Config/Include` subdirectory.
- Sitecore Azure edits the `ConnectionStrings` section of `web.config`. It removes unnecessary connection strings and adds new ones based on databases which are stored in the `DeploymentItem/Databases/ settings`. If you want add additional connection strings it can be done with `WebConfigPatch`.
- Sitecore Azure copies license information from the local installation to the content delivery package. If you need a different license file in the content delivery installation, add this file into the `"OverrideSourcesFolder\AppData"` subfolder.
- Sitecore Azure modifies the `web.config` file in the Content Delivery package to configure or eliminate databases, indexes, archives, agents, sites, watchers, and other features.

1.3.4 How Do I Use My Own Domain Name?

To use your own domain name you must create a CNAME record with your internet registrar. Using the tool provided by your internet registrar, create an entry that maps the desired alias (such as "www") for your domain to the host name that was assigned when your Windows Azure application was created (such as "mydomain.cloudapp.net"). Then use the tool provided by your internet registrar to forward the root domain to the subdomain that you have mapped to your Windows Azure host name.

CNAME records must always be mapped to a specific subdomain, you cannot map to the root domain itself or do wildcard mappings. You can find an article that describes this procedure in more detail at <http://blog.smarx.com/posts/custom-domain-names-in-windows-azure>.

Never map directly to a Virtual IP address (VIP) because in certain instances Windows Azure may change the underlying VIP.

1.3.5 How to Encrypt Your Package

The package for Sitecore Azure is encrypted by default. If you want to disable encryption set `Azure.Package.NoEncryptPackage` setting to `true`. This setting is found in the `/App_Config/Include/Sitecore.Azure.config` file.

```
<setting name="Azure.Package.NoEncryptPackage" value="true" />
```

1.3.6 How to Add AppFabric Caching

Basic information about Windows Azure Caching Service can be found at:

<http://msdn.microsoft.com/en-us/library/windowsazure/gg278356.aspx>

Windows Azure AppFabric Caching enables you to create a cache in the cloud to be used from your applications or services. This includes ASP.NET session state and output caching. For a more detailed description of this Windows Azure feature see <http://msdn.microsoft.com/en-gb/magazine/gg983488.aspx>

These steps are required for enabling your Sitecore solution to use AppFabric Caching:

- Set your Application Pool to .Net Framework version 4.0.
- Add Assemblies for AppFabric caching in to WebSite/Bin folder:

Microsoft.ApplicationServer.Caching.Client.dll

Microsoft.ApplicationServer.Caching.Core.dll

Microsoft.Web.DistributedCache.dll

Microsoft.WindowsFabric.Common.dll

Microsoft.WindowsFabric.Data.Common.dll

You can get these assemblies from \Program Files\Windows Azure SDK\V1.6\Cache.

- Start new deployment and click "More options" link. Navigate to Deployment item and in WebConfigPatch field and add following transformation under xsl:stylesheet tag.

```
<xsl:template match="configuration/configSections">
<configSections>
<xsl:copy-of select="*" />
<section name="dataCacheClients" type="Microsoft.ApplicationServer.Caching
.DataCacheClientsSection, Microsoft.ApplicationServer.Caching.Core"
allowLocation="true" allowDefinition="Everywhere"/>
</configSections>
<dataCacheClients>
<dataCacheClient name="default">
<hosts>
<host name="YOURCACHE.cache.windows.net" cachePort="PORTNUMBER"/>
</hosts>
<securityProperties mode="Message">
<messageSecurity authorizationInfo="YOURTOKEN">
</messageSecurity>
</securityProperties>
</dataCacheClient>
</dataCacheClients>
</xsl:template>

<xsl:template match="system.web/sessionState">
<sessionState mode="Custom" customProvider="AppFabricCacheSessionStoreProvider">
<providers>
<add name="AppFabricCacheSessionStoreProvider" type="Microsoft.Web.DistributedCache
.DistributedCacheSessionStateStoreProvider, Microsoft.Web.DistributedCache"
cacheName="default" sharedId="SharedApp" useBlobMode="false" />
</providers>
</sessionState>
</xsl:template>
```

1.3.7 How to Enable SSL in a Sitecore Azure Project

Basic information about using SSL with Microsoft Windows Azure can be found at:

<http://msdn.microsoft.com/en-us/library/windowsazure/ff795779.aspx>.

Sitecore settings that control SSL are found in the Service Configuration and Service Definition fields of the Deployment item. After your initial installation edit the following sections in these fields to enable SSL. See section *Where does Sitecore Azure Store Configuration?*

1.3.8 How to Enable Remote Desktop Access to Webrole Instances

Basic information about enabling Remote Desktop Access to Webrole Instances can be found at: <http://msdn.microsoft.com/en-us/library/windowsazure/hh124130.aspx>

Sitecore settings that control Remote Desktop Access to Webrole instances are found in the Service Configuration and Service Definition fields of the Deployment item. For more information see the section *Where does Sitecore Azure Store Configuration?*

Enabling Remote Desktop Access to a Webrole Instance can be done after creating a Sitecore Azure content delivery environment and adding a new certificate to the Certificate Store. Therefore a Certificate file with a private key must be uploaded to a hosted service.

Note

It is necessary to upgrade the Sitecore Azure content delivery environment after changing Configuration and Service Definition fields.

Basic information about creating a Certificate for a Role can be found at: <http://msdn.microsoft.com/en-us/library/gg432987.aspx>

Basic information about adding a new Certificate to the Certificate Store can be found at: <http://msdn.microsoft.com/en-us/library/gg465712.aspx>

Edit the following sections in these files to enable Remote Desktop Access:

ServiceConfiguration

```
<Role name="WorkerRole">
  <Instances count="5" />
  <ConfigurationSettings>
    . . .
    <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.Enabled" value="true" />
    <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountUsername"
      value="username" />
    <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.
      AccountEncryptedPassword" value="REDACTED" />
    <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountExpiration"
      value="2011-01-27T23:59:59.0000000-05:00" />
    <Setting name="Microsoft.WindowsAzure.Plugins.RemoteForwarder.Enabled" value="true" />
  </ConfigurationSettings>

  <Certificates>
    <Certificate name="Microsoft.WindowsAzure.Plugins.RemoteAccess.PasswordEncryption"
      thumbprint="REDACTED" thumbprintAlgorithm="sha1" />
  </Certificates>
</Role>
```

ServiceDefinition

```
<ServiceDefinition name="AzureAtHome" upgradeDomainCount="2"
  xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceDefinition">
  <WebRole name="WebRole">
    . . .
    <Imports>
      <Import moduleName="RemoteAccess" />
    </Imports>
  </WebRole>
```

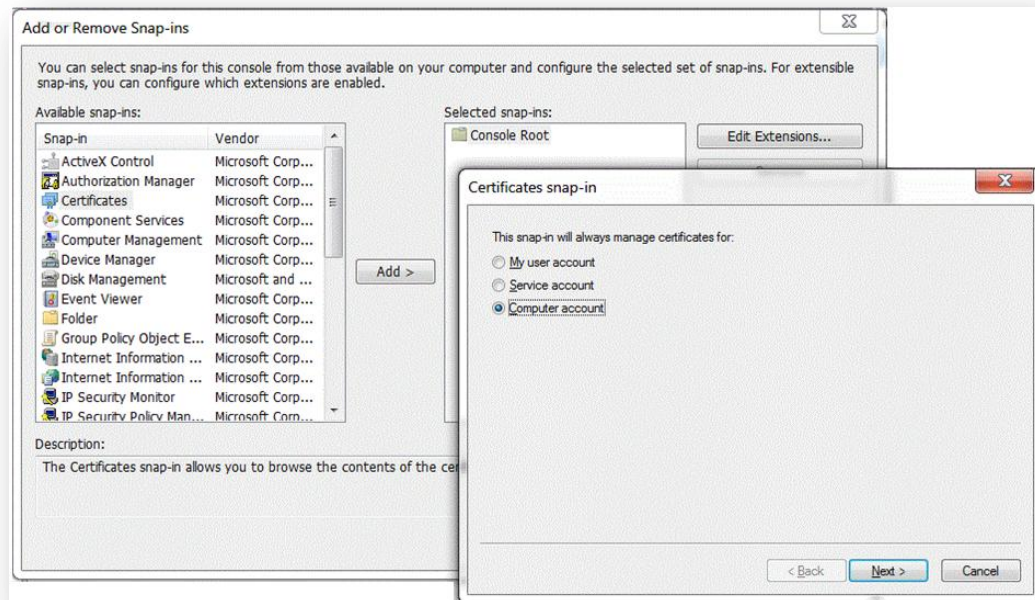
For more information about using Remote Desktop Access with Microsoft Windows Azure see <http://blogs.msdn.com/b/jimoneil/archive/2010/12/29/azure-home-part-13-remote-desktop-configuration.aspx>.

1.3.9 How to Add or Replace Management Certificates

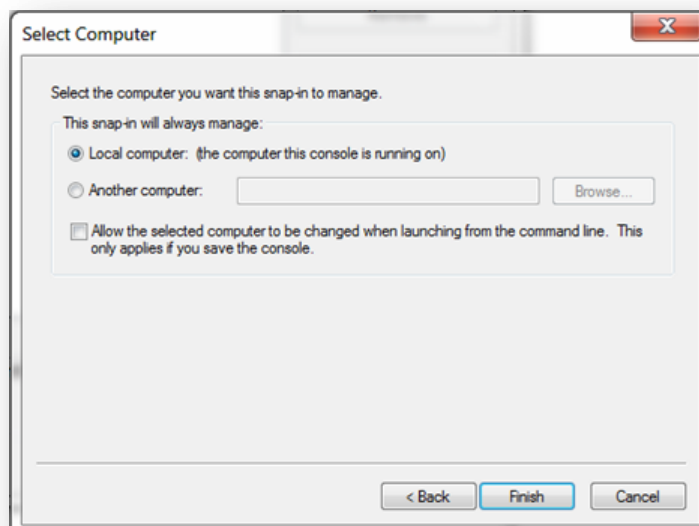
To grant the account that is used to run Application pool required security rights to the certificate:

1. Run **Microsoft Management Console**.

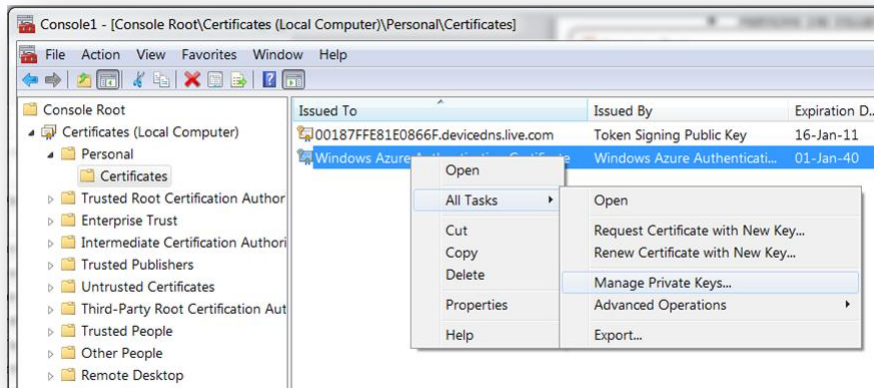
2. In the **File** menu, click **Add/Remove Snap-in**.
3. In the **Available snap-ins** list, select **Certificates** and click **Add**.
4. In the **Certificates snap-in** dialog box, select the **Computer Account** option, and then click **Next**:



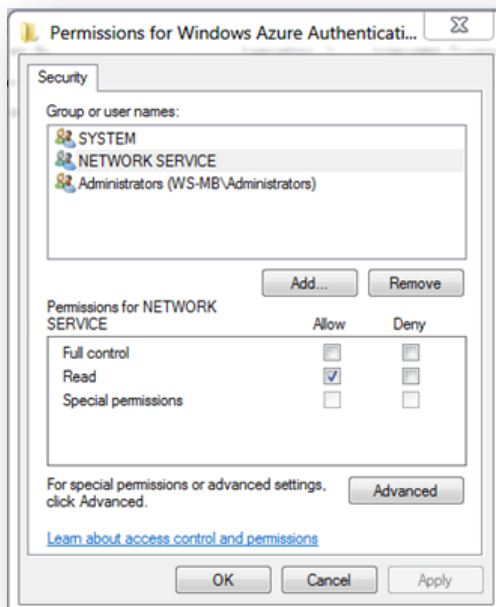
5. In the **Select Computer** dialog box, select the computer that you want to manage.



- In **Microsoft Management Console**, select **Windows Azure Authentication Certificate**, right click it, and select **All tasks — Manage Private keys**.



- Add the account that is used to run the corresponding application pool and grant it *Read* permission:



Note
 In this example the NETWORK SERVICE account is used as an example, this may vary on different configurations. You can learn more about Application Pool Identities at <http://learn.iis.net/page.aspx/624/application-pool-identities/>.

1.3.10 Environment File Deployment Naming Convention

When your Sitecore Azure Environment File is created a deployment name will be generated. The deployment name must be globally unique and may be used for multiple purposes in an Azure implementation. The deployment name will be assigned using the following pattern:

```
[projectname][environmenttype][location]sc[first 3 chars of subscription id]
```

For example a deployment name of `myprojDevNcusc91a` would indicate a project with the name of “myproj” running as a development environment, using the North Central US hosting location.

There are four types of Sitecore Azure environments:

- Evaluation
- Development
- Test
- Production

1.3.11 How to Configure Firewall Rules

When configuring firewall rules, allow access for the following common servers and ports.

For Microsoft datacenters:

<https://management.database.windows.net:8443>

<https://management.core.windows.net> (443)

<https://blob.core.windows.net> (443)

In addition, depending on which deployment location you are using port 1433 needs to be opened on the following hosts:

- **West Europe** — `data.am2-1.database.windows.net` [94.245.97.22]
- **North Europe** — `data.db3-1.database.windows.net` [94.245.114.93]
- **North Central US** — `data.ch1-1.database.windows.net` [207.46.203.22]
- **East Asia** — `data.hkn1-1.database.windows.net` [111.221.69.22]
- **South East Asia** — `data.sgp1-1.database.windows.net` [207.46.63.13]
- **South Central US** — `data.sn1-1.database.windows.net` [65.55.74.144]

1.3.12 How to Create a Database Account for Sitecore Azure on SQL Server

Create a database account for Sitecore CMS databases on SQL Server as described in the Installation Guide for your Sitecore version and add the permissions necessary for Sitecore Azure. You can find the Sitecore installation guides at <http://sdn.sitecore.net/Products/Sitecore%20V5/Sitecore%20CMS%206/Installation.aspx>.

To create a Database Account for Sitecore Azure on SQL Server:

1. In MS SQL Server Management Studio, in the Object Explorer window, expand the Databases node.
2. For both the Sitecore Core and Master databases, right click on the database name and select the Properties submenu. The Database Properties dialog box appears.
3. In Database Properties, select the Permission page.
4. In the Users or roles field, select your SQL user and in the Permission for `UserName` field, grant the View Definition permission.

1.3.13 How to Create a Database Account for Sitecore Azure on SQL Azure Server

To create a Sitecore Azure content delivery environment it is necessary to use the default SQL Azure admin account, which is embedded in the environment file. This is needed to create a SQL Azure database server and databases.

If the SQL Azure database server is created with corresponding databases or the entire Sitecore Azure content delivery environment is created, you can use a new SQL Azure user account with a minimal set of permission instead of the SQL Azure admin user account.

For information about where Sitecore Azure stores database project configuration information for deployment to SQL Azure, see the section *Where does Sitecore Azure Store Configuration?*

Note

It is necessary to upgrade the Sitecore Azure content delivery environment after changing the SQL Azure user account if it has already been deployed.

To Create a Database Account for Sitecore Azure on SQL Azure Server:

Create a System Administrator (`sa`) SQL Azure login and a SQL Azure user in the Core and Web databases on the SQL Azure database server. Basic information about managing database and logins in SQL Azure can be found at:

<http://msdn.microsoft.com/en-us/library/windowsazure/ee336235.aspx#DatabasePerms>

Add `'db_datareader'` and `'db_datawriter'` roles on the Core and Web databases for the SQL Azure user.

Grant `'Execute'` and `'VIEW DATABASE STATE'` permissions on the Core and Web databases for the SQL Azure user. Basic information about granting database permission can be found at:

<http://msdn.microsoft.com/en-us/library/windowsazure/ee336285.aspx>

1.4 Known Issues, Considerations, and Troubleshooting

This section describes known issues with Windows Azure, SQL Azure, and Sitecore Azure.

1.4.1 ASP.NET Considerations on Windows Azure

Consider the following for any ASP.NET application such as Sitecore on Windows Azure.

- Eliminate use of ASP.NET session or consider `TableStorageSessionStateProvider` for session management.
- Remote access to the Windows Azure environment is extremely limited. You cannot attach a debugger to the Windows Azure environment. You cannot directly access the file system of a Windows Azure environment, including log files.
- Before moving to the cloud, consider any integration within the solution to internal applications.
- Use configuration files rather than Windows IIS Manager.
- You cannot deploy an individual file to Windows Azure. You must deploy the entire document root.

1.4.2 SQL Azure Considerations

Consider the following when using SQL Azure with applications such as Sitecore.

- When accessing an SQL Azure database from a remote location, performance depends on network conditions and other factors.
- Use an encrypted connection over port 1433 when accessing SQL Azure remotely (outbound access from the content management environment).

For information about SQL Azure security, see <http://msdn.microsoft.com/en-us/library/ff394108.aspx>.

- SQL Azure licensing limits the maximum database size.

For information about SQL Azure sizing and licensing, see <http://msdn.microsoft.com/en-us/library/ee336245.aspx#dcasl>.

- If SQL Server 2008R2 is installed it must be with SP1.

1.4.3 Sitecore Azure Considerations

When you use Sitecore Azure:

- You cannot use the automated installer to install Sitecore on Windows Azure. You can use the automated installer to create a local Sitecore instance, and then use Sitecore Azure to replicate that environment to the cloud.
- Sitecore Azure *only* supports SQL Azure databases.
- Sitecore Azure does not support file media.
- Windows Azure uses certificates to exchange files securely over HTTP. Sitecore Azure Environment files are restricted to license. SQL Azure uses IP address restrictions for database security. Work with Sitecore Azure services if you plan to move your content management instance from one machine or location to another.
- Sitecore has not tested modules including the Online Marketing Suite (OMS 1.0) and Digital Marketing Suite (DMS 2.0) on Windows Azure.

1.4.4 Troubleshooting Sitecore Azure

General troubleshooting steps for Sitecore Azure include:

- Restart ASP.NET.
- Clear the browser cache.
- Investigate the Sitecore logs in the content management environment for errors.
- Remove the instances from the environment and deploy again.
- After deployment, configure the `$dataFolder/AzurePackages\ (xxx) Deployment` subdirectory as the document root of an IIS web site, and diagnose that site.
- Temporarily set the `mode` attribute of the `/configuration/system.web/customErrors` element in the `web.config` file to `Off`.

Warning

Setting the `mode` attribute to `Off` can expose detailed error information in the Sitecore Azure environment. Consider other techniques to protect the application while `mode` is `Off`. Remember to set `mode` to `On` or `RemoteOnly` after diagnosis, and remember to deploy that change.

1.4.5 Creating a Self-Signed Certificate

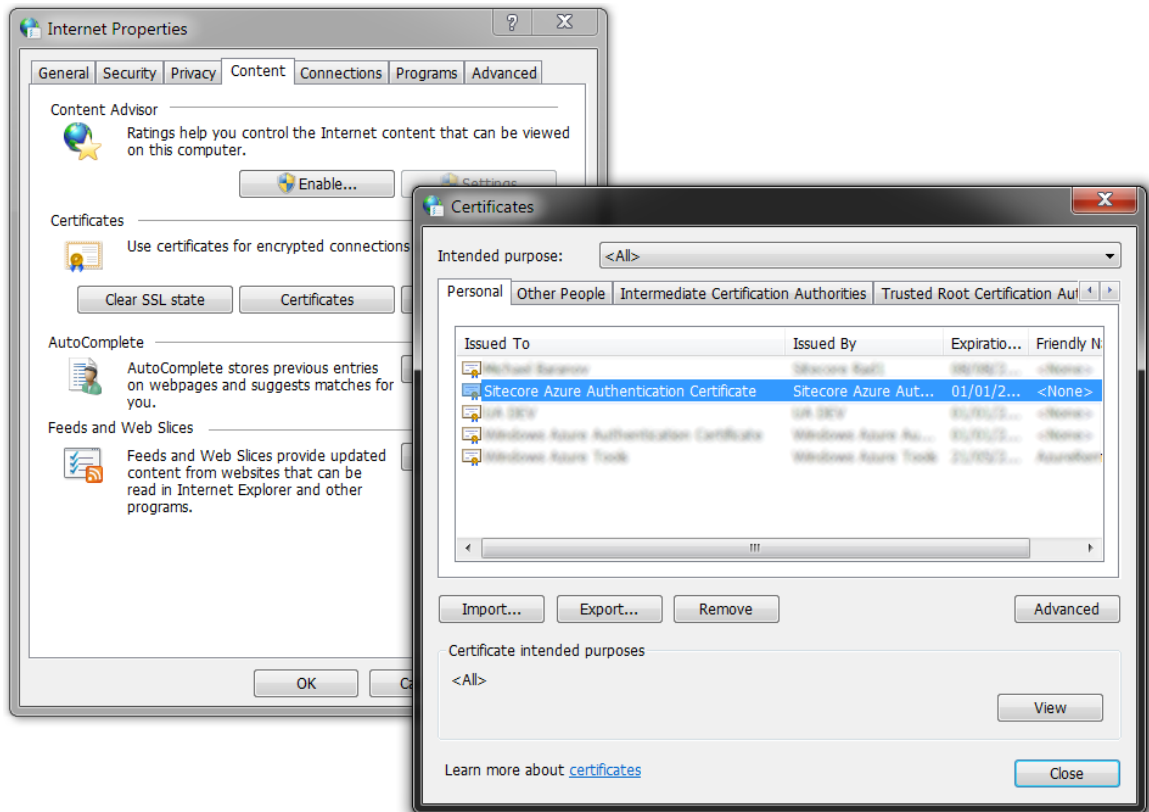
If you do not have a certificate from a Certificate Authority already installed on your web server, you can create a self-signed certificate for development and testing purposes.

1. Generate the self-signed certificate by using the `MakeCert.exe` tool. The following is an example of a command to create a certificate. This command will create a new key with private and public parts. The public key will be placed in file system (`certificate.cer`). The private key will be installed into Windows certificate storage.

```
makecert -r -pe -a sha1 -n "CN=Sitecore Azure Authentication Certificate"  
-ss My -len 2048 -sp "Microsoft Enhanced RSA and AES Cryptographic Provider"  
-sy 24 "certificate.cer" -sr CurrentUser
```

2. Locate the public key (`certificate.cer`) generated in the previous step and upload it to the Windows Azure Management Certificate store.

- Export the private key certificate as a .pfx file. This can be done from Control Panel, Internet Options or with use of MMC -> Add/Remove Snap-in -> Certificates.



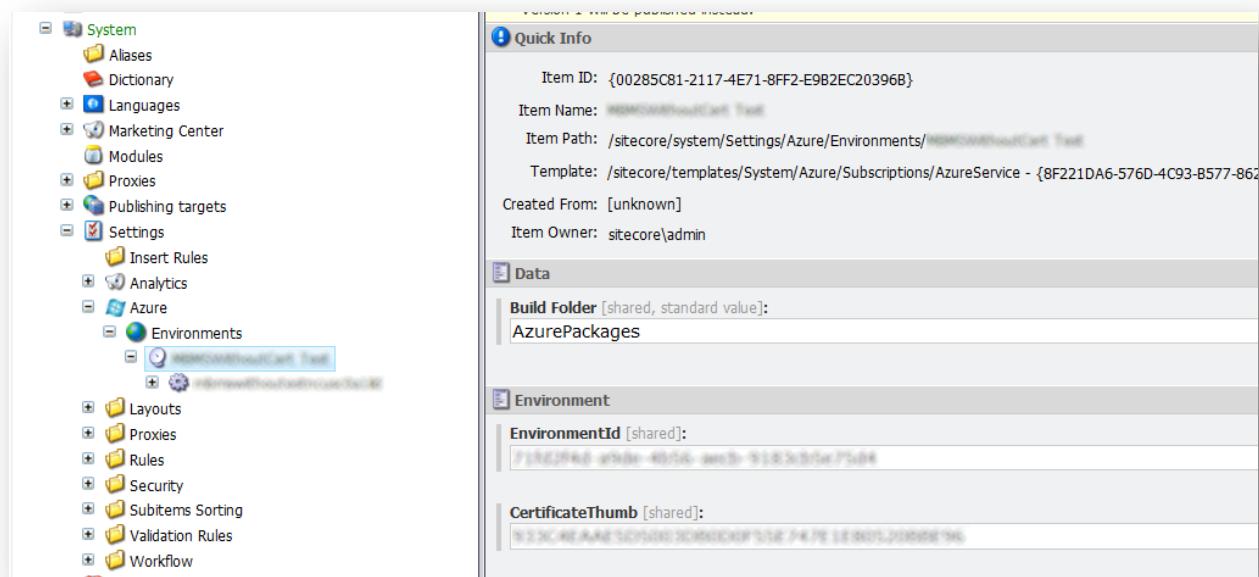
For more information about the `Makecert.exe` tool, see <http://msdn.microsoft.com/en-us/library/bfskty3.aspx>.

1.4.6 How to Replace a Certificate

If you need to replace an existing certificate for example., because of certificate compromise, you need to:

- Upload a new public key into the Windows Azure Management Certificate store.

2. Delete the contents of the *CertificateThumb* field in the `/sitecore/system/Settings/Azure/Environments/<Project Name>` item.



The next time you create a new content delivery instance Sitecore Azure will prompt you to upload the new certificate.

Important

The certificate will be installed under the account that is used for the AppPool. Sitecore Azure does not store the `.pfx` file or the password. This information is only used during certificate installation.