



Contents

1	INTRODUCTION	3
1.1 1.2	Who should read this guide Getting started	3 3
2	PRE-INSTALLATION AND INSTALLATION STEPS	5
2.1	Installation	5
3	CONFIGURING MEDIA FILE SYNCHRONIZATION	7
3.1 3.2 3.3 3.4	Creating a container in Azure blob storage Getting the blob connection string Specifying the blob connection string in Web.config Configuring the media file provider in Composite.config	7 7 8 8
4	CONFIGURING DATA CHANGE SYNCHRONIZATION	10
4.1 4.2 4.3 4.4	Creating a service bus on Microsoft Azure Getting the connection string for the service bus Specifying the service bus connection string in Web.config Configuring the data provider in Web.config	10 10 11 11
5	VERIFYING THE CORRECT CONFIGURATION OF SYNCHRONIZA	TION 13



1 Introduction

The SQL Azure Scale Out add-on allows you to scale out C1 CMS websites deployed on Microsoft Azure.

Microsoft Azure 🛛 🗸	Subscriptions	7 ⊕
	deployment	
	ASHBOARD MONITOR CONFIGURE SCALE INSTANCES LINKED RESOURCES CERTIFICATES	
deployment	PRODUCTION STAGING	
•	TRANSITIONING The deployment is transitioning. 3 Instances: 2 Starting, 1 Running	
<u></u>	NAME	
DB	WebRole_IN_0 🗸 Running	
	WebRole_IN_1 * Starting VM	
	WebRole_IN_2 * Starting VM	
\odot		
2		
÷		
<u>~</u>		
*		
11		
PH4		

Figure 1: Multiple instance of C1 CMS website deployment on Microsoft Azure

The scaled-out setup works for multiple-instance SQL-based C1 CMS deployments where it handles media and data change synchronization.

As a result, content and media file changes are immediately updated on all website instances.

It is an Azure-cloud equivalent of the <u>Load Balancing</u> add-on for on-premise deployments.

1.1 Who should read this guide

The guide is intended for a technical person who is familiar with Microsoft Azure and the network load balancing technologies and capable of setting up entities such as storage accounts or service buses on Microsoft Azure.

We expect that this person has an SQL-based C1 CMS website running or is able to migrate an XML-based website to an SQL-based data store.

This person should have access to the System perspective in the CMS Console to install add-ons, migrate the website and make backups if needed, and check the logs. Please note that he or she should also have access to a Microsoft Azure account.

1.2 Getting started

Before you install the add-on, you need to take a few pre-installation steps.

After you've installed the add-on, make sure to:



- 1. Configure media files synchronization.
- 2. Configure data changes synchronization.

Once you've configured the synchronization within your scaled-out setup, you may want to <u>check the logs for synchronization errors</u> to make sure that everything is working properly.



2 Pre-Installation and Installation Steps

Before you install and configure the add-on, make sure to:

- 1. Use Composite C1 (now C1 CMS) version 4.2 Update 2 or later.
- 2. <u>Deploy the website on Microsoft Azure</u>.
- 3. Convert the XML-based website into an SQL-based one.

You may want to consider:

- Using an <u>C1 CMS Azure service package</u> with the "Small" or larger VM size for the web role being deployed.
- <u>Allowing write-backs</u> on the website to be able to work in the CMS Console of the website: <Runtime writeback="true" websiteType="composite-c1" />
- Starting with a 1-instance deployment. (You may increase the number of instances in the Microsoft Azure management portal after you have configured your scaledout solution.)

2.1 Installation

Installing the SQL Azure ScaleOut add-on is no different from installing any other CMS add-on:

- 1. Log into the CMS Console as an administrator.
- 2. From the "System" perspective, expand "Packages", "Available Packages", and "Composite.Azure" and select " Composite.Azure.ScaleOut"

51	SYSTEM
≡	Q PACKAGE INFO
	∼ 🏯 Packages
	– 🗸 🖿 Available Packages
	- > 🖿 CkSoftware
	– > 🖿 Composite.AspNet
Ē	– 🗸 🚞 Composite.Azure
	— 🔂 Composite.Azure.Publisher
ß	Composite.Azure.ScaleOut
ν	- > Composite.Commerce

Figure 2: The SQL Azure ScaleOut add-on

- 3. Click "Package Info" on the toolbar.
- 4. In the "Package Info" view, click "Install".



5. Complete the wizard.

							admin 🗸	\$ \$
ш		SYSTEM	Q SCALEOUT (COMPOS 🛽	Q SCALEOUT (COMPOS 🛞				
≡	PAC	CKAGE INFO	TION INFO			<u>4</u>	INSTALL	🔗 READ MORE
	P	PACKAGE INFO			FRE	E TRIAL INFO		
B		Name				Trial information		
8		Composite.Azure.ScaleC	Out			This is a commercial package, available for free in the trial period. When the trial period has expired functionality may be degraded, unless you choose to purchase a license.		
6 5		The Composite.Azun that allow visitors to add instance Azure deploym Requires SQL data p Uses ServiceBus to n flushing cached data; us Arthing changes	e.Scale package is applicable for we d/edit data, while scaling out using i ients. rovider, or a similar custom provide notify instances about changes in da ses Blob Storage for synchronizing N	ebsites multi r. ta, thus Aedia		Free trial period (days) 30		
ŕ		Author				Price (USD)		
ইট		Composite A/S				1950		

Figure 3: Installing the SQL Azure ScaleOut add-on

Now you are ready to configure the work of the installed add-on by setting up <u>media</u> and <u>data</u> change synchronization.



Configuring Media File Synchronization 3

To configure media files synchronization:

- 1. Create a container in Azure blob storage
- 2. Get the connection string to the blob storage
- 3. Specify the blob connection string in Web.config
- 4. Configure the media file provider in Composite.config

3.1 Creating a container in Azure blob storage

For synchronization, you first need to create a container in Azure blob storage. This is where the scaled-out C1 CMS website will store its media files (instead of the standard place at ~/App_Data/Media):

- 1. Log into the Microsoft Azure management portal.
- Choose an existing blob storage account or, if needed, create a new one.
 Add a storage container (for example, "c1mediafiles") where synchronized media files will be stored.
- 4. Make sure the container has the "Private" access.

New container

NAME		
c1mediafiles		
ACCESS 🕜		
Private		•

Figure 4: Creating a new container on Microsoft Azure

3.2 Getting the blob connection string

In the following steps you'll configure your website to synchronize media file. For this, you need to know the connection string to the blob storage where you've created a dedicated container (as described in the step above).

- 1. In the Microsoft Azure management portal, select the blob storage with the container.
- 2. Click "Manage Keys".
- 3. Make a note of, or copy, the storage account name and storage access key.



Manage Access Keys

When you regenerate your storage access keys, you need to update any virtual machines, media services, or applications that access this storage account to use the new keys. Learn more.



Figure 5: Getting the values for the blob connection string on Microsoft Azure

3.3 Specifying the blob connection string in Web.config

Now you need to specify the blob connection string in Web.config:

- 1. Edit ~/Web.config.
- 2. Add the blob storage connection string with some name, for example, "c1media":



Listing 1: Specifying the blob connection string in Web.config

where:

- [AccountName] is the name of the blob storage account
- [AccessKey] is the access key to the blob storage account

3.4 Configuring the media file provider in Composite.config

Finally, you need to replace the default media provider in Composite.config with the Azure media provider using the name of the blob connection string you've added in Web.config and the name of container you've created in the blob storage:

- 1. Edit ~/App_Data/Composite/Composite.config.
- 2. Search for name="MediaFileDataProvider" to locate the default media provider.
- 3. Comment out this media file provider.
- 4. Add a reference to AzureMediaProvider as shown below:



```
<configuration>
  <!-- skipped -->
  <Composite.Data.Plugins.DataProviderConfiguration
defaultDynamicTypeDataProviderName="...">
    <DataProviderPlugins>
      <!-- skipped -->
      <!-- <add rootDirectory="~/App_Data/Media"
        storeId="MediaArchive"
        storeDescription="Media Archive Files"
        storeTitle="Media Archive"
type="Composite.Plugins.Data.DataProviders.MediaFileProvider.MediaFileProvi
der, Composite, Version=1.0.3037.13741, Culture=neutral,
PublicKeyToken=null"
        name="MediaFileDataProvider" /> -->
      <add rootDirectory="~/App Data/Media"
        storeId="MediaArchive"
        storeDescription="Media Archive Files"
        storeTitle="Media Archive"
type="Composite.Azure.ScaleOut.AzureMediaProvider.AzureMediaDataProvider,
Composite.Azure.ScaleOut.AzureMediaProvider"
        name="MediaFileDataProvider"
        blobStorageConnectionStringName="[BlobConnectionStringName]"
        blobContainer="[BlobContainerName]" />
    </DataProviderPlugins>
</configuration>
```

Listing 2: Configuring media file provider in Composite.config

where:

- [BlobConnectionStringName] is the name of the blob connection string added in Web.config (see above).
- [BlobContainerName] is the name of the container created in the blob storage (see above).

For example:

```
<add ...
blobStorageConnectionStringName="clmedia"
blobContainer="clmediafiles" />
```

Listing 3: Example of the media file provider configured in Composite.config



Configuring Data Change Synchronization 4

To configure synchronization of data changes, you should:

- 1. Create a service bus on Microsoft Azure (if needed)
- 2. Get the connection string for the service bus
- 3. Specify the service bus connection string in Web.config
- 4. Configure the data provider in Web.config

4.1 Creating a service bus on Microsoft Azure

To create a service bus on Microsoft Azure:

- 1. Log into the Microsoft Azure management portal.
- Create a service bus.
 Make sure that the "Messaging Tier" parameter is set to "Standard" (not "Basic")

CREATE A NAMESPACE

Add a new namespace

NAMESPACE NA	AME
mysrvbus	0
	.servicebus.windows.net
REGION	
Central US	•
TYPE 👩	
MESSAGING	NOTIFICATION HUB
MESSAGING TIE BASIC ST	

Figure 6: Creating a service bus on Microsoft Azure

Please note that when adding a service bus, the "Messaging Tier" parameter must be set to "Standard" (not "Basic") because the add-on is using the "Topics" feature of the Service Bus.

4.2 Getting the connection string for the service bus

- 1. In the Microsoft Azure management portal, select the service bus.
- 2. Click "Connection Information".
- 3. Make a note of, or copy, the service bus connection string.



Access connection information

Use this connection information to manage namespace 'mysrvbus'. You can also use authorization policies configured here to connect to all entities in this namespace.

SAS 🕜		
NAME	CONNECTION STRING	P
RootManageSharedAccessKey	Endpoint=sb://mysrvbus.servicebus.windows.net/;SharedAccessKeyName=Rootl	Manag

Figure 7: Getting the service bus connection string on Microsoft Azure

Please note that there are two types of service bus connection strings: SAS and ACS. You can use either.

4.3 Specifying the service bus connection string in Web.config

When being installed, the add-on adds two application settings to Web.config.

Listing 4: Specifying the service bus connection string in Web.config

To specify the service bus connection string:

- 1. Edit ~/Web.config.
- Set the service bus connection string in the "Composite.Azure.ScaleOut.DataCacheFlusher.ServiceBus.ConnectionString" setting. For example:

```
<add
```

```
key="Composite.Azure.ScaleOut.DataCacheFlusher.ServiceBus.ConnectionString"
value="Endpoint=sb://mysrvbus.servicebus.windows.net/;SharedAccessKeyName=R
ootManageSharedAccessKey;SharedAccessKey=gkTvceSqsUAx+okXzOG8WKCOJ9MflR/cx0
ZBwyA0QW0=" />
```

Listing 5: Example of the service bus connection string specified in Web.config

4.4 Configuring the data provider in Web.config

As you are using the default SQL data provider named "DynamicSqlDataProvider", you don't need to make any changes to the settings added by the add-on in Web.config.

• Therefore, keep the "Composite.Azure.ScaleOut.DataCacheFlusher.DataProviders" setting unchanged in Web.config.



<add key="Composite.Azure.ScaleOut.DataCacheFlusher.DataProviders" value="DynamicSqlDataProvider" />

Listing 6: The data provider configured by default in Web.config

In general, the "Composite.Azure.ScaleOut.DataCacheFlusher.DataProviders" setting contains a comma separated list of data providers for which the cache flush notification should be enabled. By default, on an SQL-based C1 CMS website, the SQL data provider named "DynamicSqlDataProvider" is used. That's why normally this setting should be kept unchanged.



5 Verifying the Correct Configuration of Synchronization

Once the add-on has been installed and synchronization of media files and data changes configured, there is need not take any special steps to ensure content change synchronization. Changes to content and media files will synchronize automatically.

However, you may want to ensure the synchronization has been configured correctly:

- 1. Log into the CMS Console of the website.
- 2. In the System perspective, open the server log.
- 3. Make sure there are no errors related to the work of the add-on and synchronization.



Figure 8: Checking for errors in the log in C1 CMS

If there are errors, please consider redoing the configuration and, if needed, reinstallation of the add-on – as described above.

