Azure Publisher - Setup Guide

2017-06-20

Orckestra, Europe
Adelgade 12, First Floor
DK-1304 Copenhagen
Phone +45 3915 7600

c1.orckestra.com

Contents

1 Introduction

The Azure Publisher add-on enables you to copy one or more C1 CMS websites from one environment, for example, an internal staging server, to an Azure blob storage.

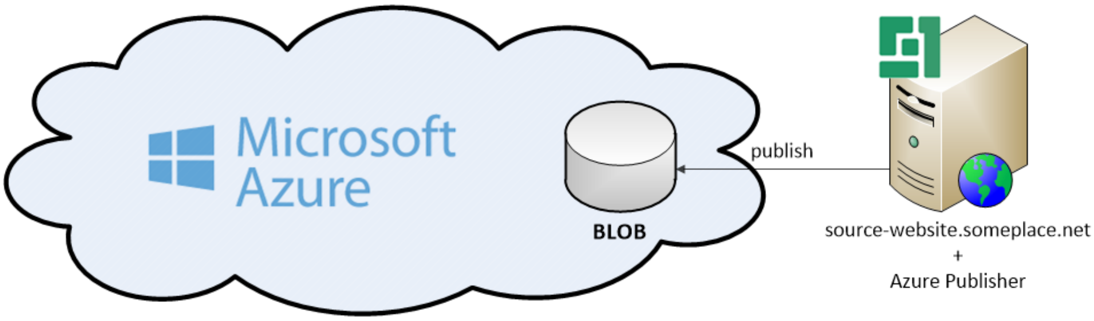


Figure 1: Copying a website to a blob storage on Microsoft Azure

Once the website is copied to the storage, it can be further synchronized to one or more live or testing servers.

Apart from copying the entire website to a blob storage, you can choose to only copy changes made since the last upload to the storage.

You can also fine-tune Azure Publisher to copy selected parts of the website or skip or overwrite selected files when copying.

1.1 Who should read this guide

The guide is intended for a technical person with access to the Microsoft Azure management portal, capable of creating a blob storage on Microsoft Azure.

This person should have a C1 CMS XML-based set up and running and know how to work in the CMS Console in general, and install C1 add-ons in particular.

Besides, access to the “Content” and “System” perspectives is required.

1.2 Getting started

Before you begin, you need to make sure that the prerequisites are in place.

You need to have a blob connection information handy as well as have an XML-based website up and running.

Then you can go on and:

1. Install Azure Publisher
2. Configure Azure Publisher

The further steps of copying the website to the blob storage and then to one or more target servers are not covered in this guide.

Please see the “Azure Publisher User Guide” for information about using Azure Publisher to copy websites to a blob storage.

Please see the “Azure Server Setup Guide” for information about copying from a blob storage to one or more target servers.

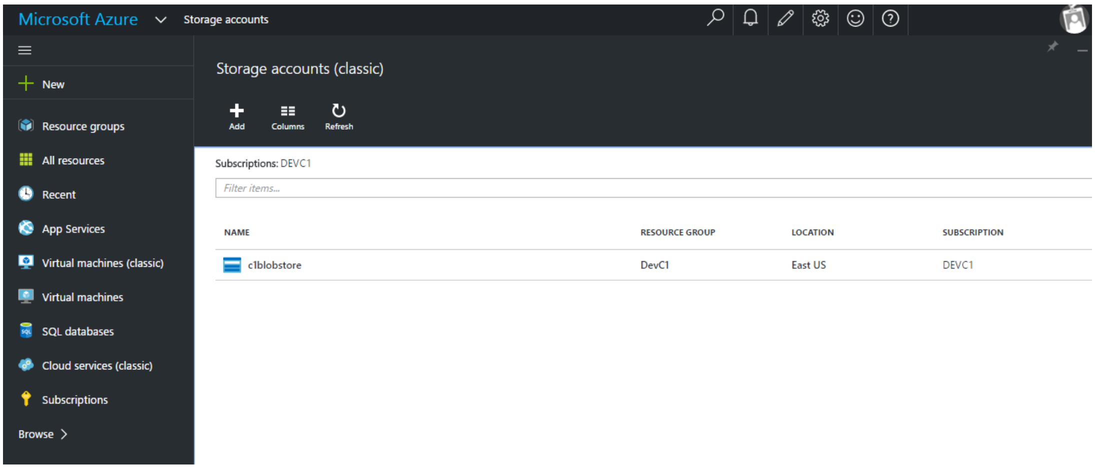
2 Prerequisites

Make sure you have these prerequisites in place:

- A blob storage on Microsoft Azure
- A source C1 CMS website

2.1 Azure blob storage

You should have an Azure blob storage (classic) where you will upload your “source” website or changes on it.



NAME	RESOURCE GROUP	LOCATION	SUBSCRIPTION
c1blobstore	DevC1	East US	DEVC1

Figure 2: A blob storage on Microsoft Azure

If you do not have a storage on Microsoft Azure, create one:

1. Log in to your Microsoft Azure Portal.
2. Select **New** / **Storage** / **Storage account**.

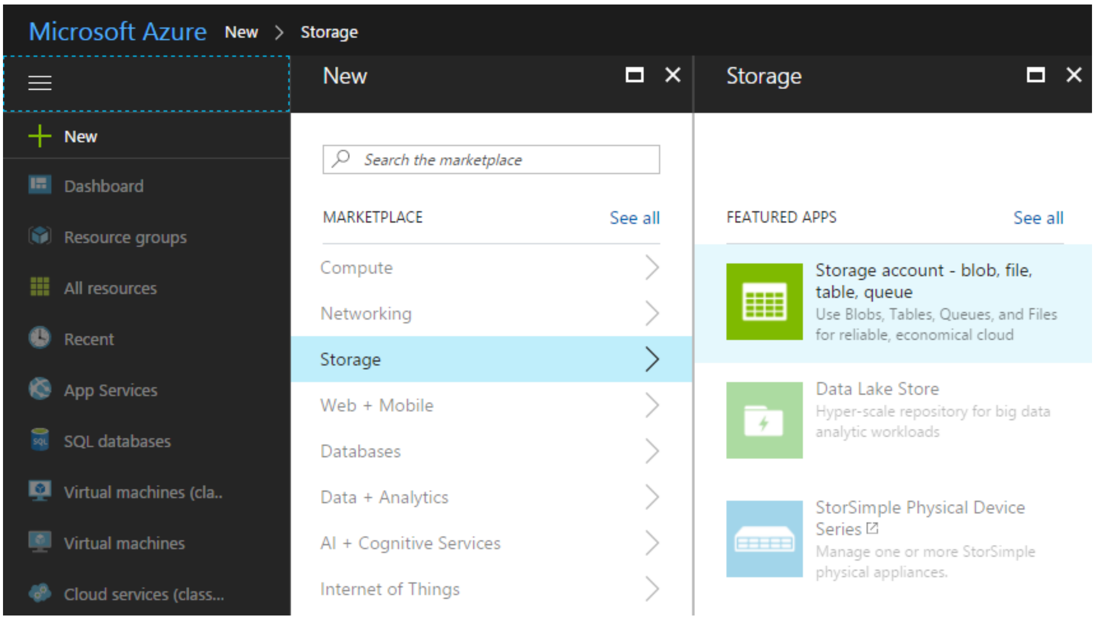
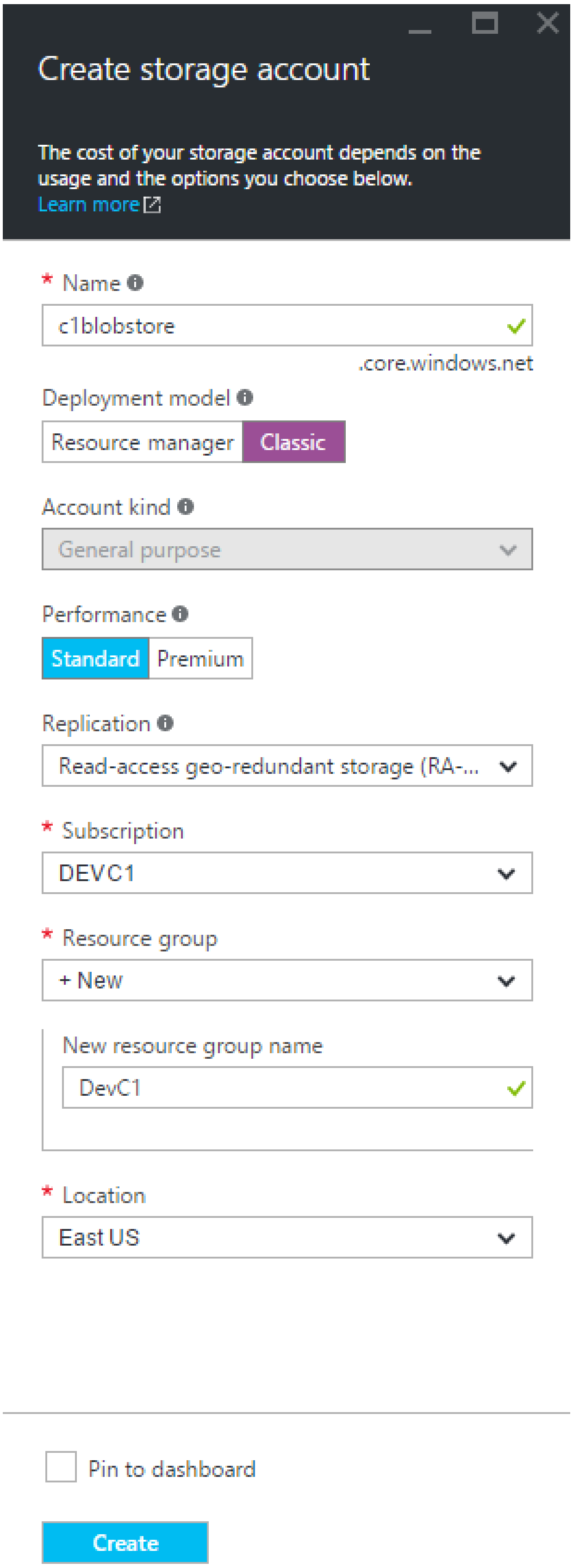


Figure 3: Creating a new storage account on Microsoft Azure

3. Select **Classic** for the deployment model.

4. Enter a name for your storage account and select other options where necessary: **Type**, **Subscription**, **Resource Group**, **Location**.
5. Click **Create**.



Create storage account

The cost of your storage account depends on the usage and the options you choose below.
Learn more

* Name
c1blobstore ✓
.core.windows.net

Deployment model
Resource manager | Classic

Account kind
General purpose

Performance
Standard | Premium

Replication
Read-access geo-redundant storage (RA-...

* Subscription
DEVC1

* Resource group
+ New

New resource group name
DevC1 ✓

* Location
East US

□ Pin to dashboard

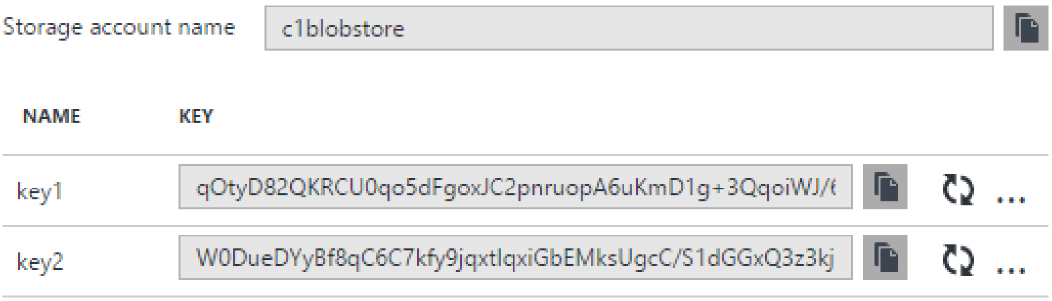
Create

Figure 4: Filling out configuration parameters of a blob storage

2.1.1 Blob connection information

Next, copy or make a note of, the storage account's *name* and *access key (key1)*:

- **All Resources** / **[your storage account]** / **Access Keys**



Storage account name: c1blobstore

NAME	KEY
key1	qOtyD82QKRCU0qo5dFgoxJC2pnruopA6uKmD1g+3QqoiWJ/
key2	W0DueDYyBf8qC6C7kfy9jqxtlqxiGbEMksUgcC/S1dGGxQ3z3kj

Figure 5: Getting the blob storage name and access key

When configuring Azure Publisher you will need to specify the storage account name and primary access key for the blob connection.

It is a “recipient” endpoint (target). Here, you will upload the “source”.

2.2 Source website

This is a C1 CMS XML-based website. (The SQL-based website is not covered in this guide.)

We recommend using C1 CMS version 5.0 or later.

You can deploy it wherever it is convenient for you. It might be a local website running on WebMatrix, or an online instance on Microsoft Azure or your ISP.

It is a “sender” endpoint (source). Changes made here will be pushed to a blob storage by Azure Publisher.

The source website can be considered as a “staging” website while the website on a target server - a “production” or live website (especially in a single-deployment scenario).

This website can also be occasionally called “local”, which might be a little misleading, since you are not limited to where you host this website: on your local or some remote machine.

3 Installing Azure Publisher

To install the Azure Publisher add-on on the source website:

1. Log in to the CMS Console.
2. From the **System** perspective, expand **Packages**, **Available Packages, Composite.Azure**.
3. Select **Composite.Azure.Publisher** and click **Install**.
4. Complete the add-on's installation wizard.

Once the add-on has been installed, the CMS Console will reload and the "Azure Publisher" element will appear in the Content perspective.

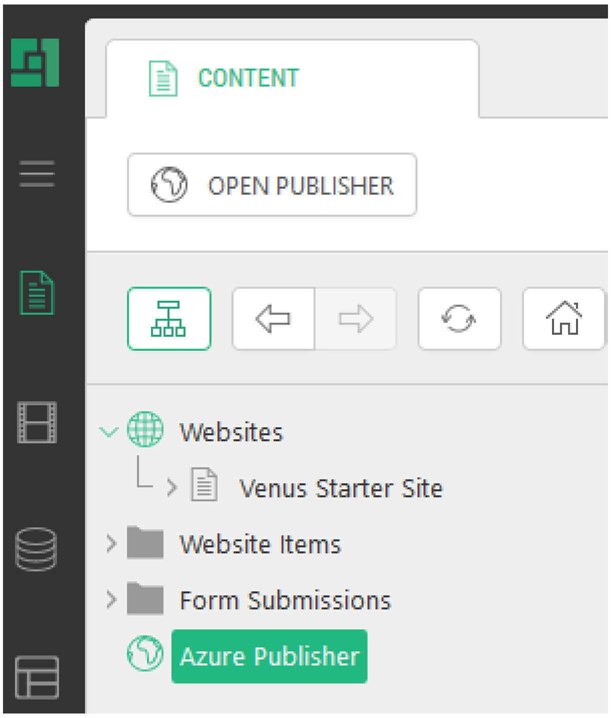


Figure 6: Azure Publisher in the Content perspective

4 Configuring Azure Publisher

Configuring Azure Publisher includes:

1. Setting up connection to a blob storage
2. Setting up one or more publishing profiles

As Azure Publisher comes with predefined profiles, you may want to skip Step 2 altogether.

4.1 Configuring connection to a blob storage

As Azure Publisher uploads a staging website to a dedicated container ("website container") in an Azure blob storage, you need to configure conection using the blob connection information that includes:

- the storage account name
- its access key

To configure the connection:

1. In the **Content** perspective, right-click **Azure Publisher** and click **Configure** in the shortcut menu to open the **Configuration** view.

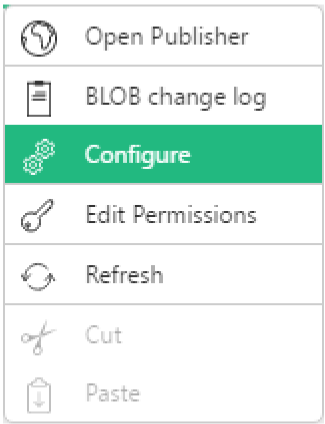
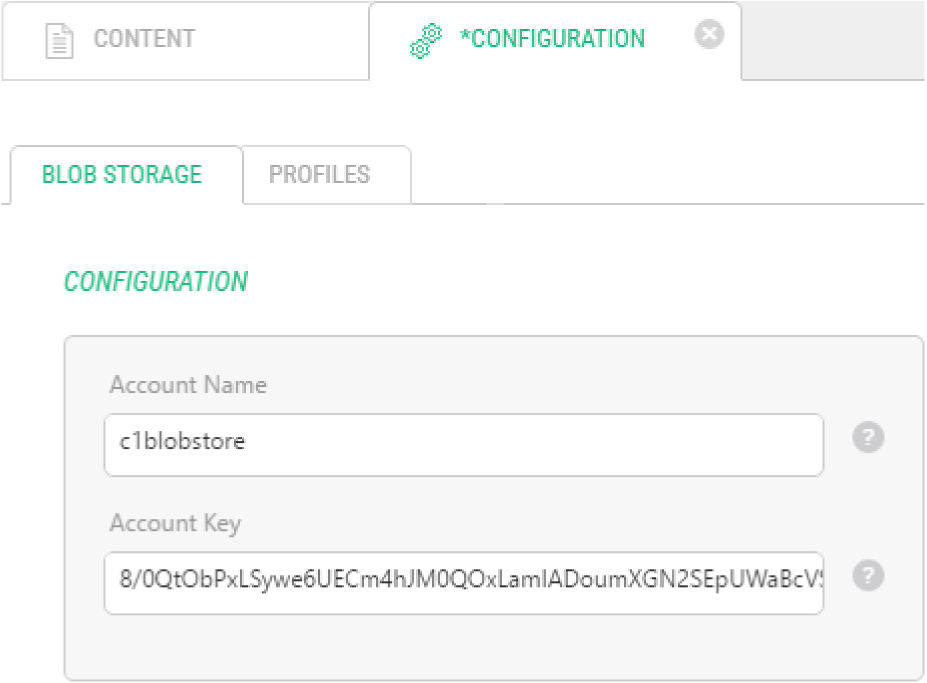


Figure 7: Selecting Configure from the menu

2. On the "**Blob Storage**" tab, within the "**Configuration**" group box, fill out these fields:
   - **Account Name**: Your Azure storage account's name, for example, *c1blobstore*
   - **Account Key**: Your Azure storage account's access key, for example, *f4uLInNSRj9HiP9zP7FeN8dLRusthmdQggufw+yLvmaMLZzyslJzPOG65YfrIPuoKmu7rKEJmszzqHzQpKZx1A==*
3. Click **Save**.



CONFIGURATION

Account Name: c1blobstore

Account Key: 8/0QtObPxLSywe6UECm4hJM0QOxLamIADoumXGN2SEpUWaBcV

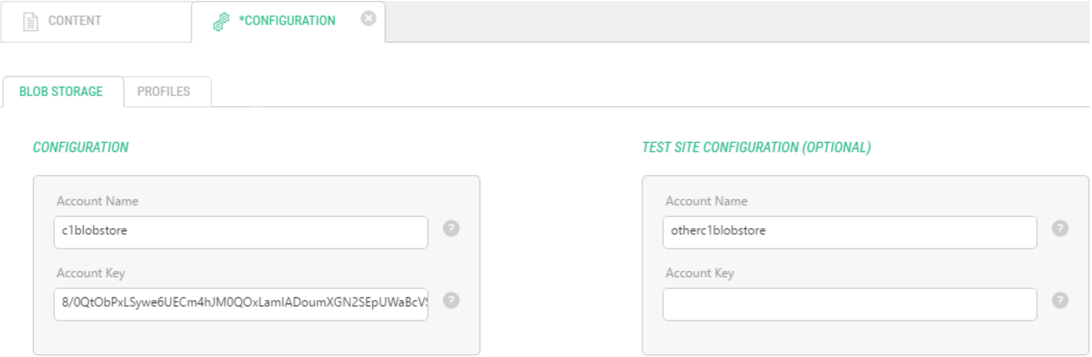
Figure 8: Configuring Azure Publisher

Now continue to the ”Profiles” tab to set up one or more profiles for publishing.

4.1.1 Configuring a test site

If you are using another blob storage for test purposes, you can upload to that storage, too.

1. In the **Content** perspective, right-click **Azure Publisher** and click **Configure** in the shortcut menu to open the **Configuration** view.
2. On the “**Blob Storage**” tab, within the "**Test Site Configuration (optional)**" group box, fill out the fields as you did for the main configuration using the values for your test website this time.
3. Click **Save**.



CONFIGURATION

Account Name: c1blobstore

Account Key: 8/0QtObPxLSywe6UECm4hJM0QOxLamIADoumXGN2SEpUWaBcV

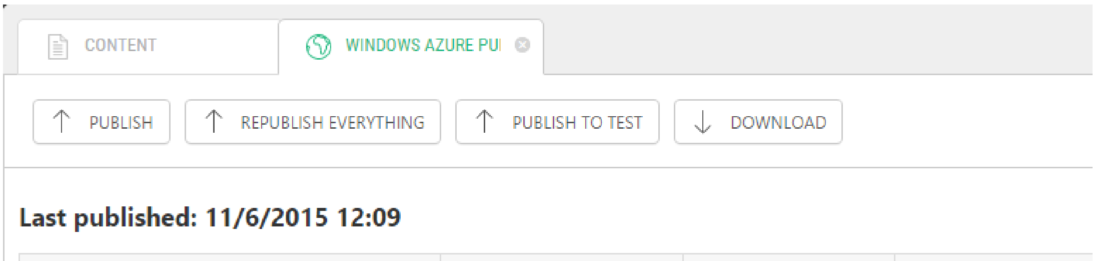
TEST SITE CONFIGURATION (OPTIONAL)

Account Name: otherc1blobstore

Account Key:

Figure 9: Configuring Azure Publisher for a test website

If the test website is configured, a new button – “Publish to Test” – will appear in the Advanced Publisher view so that you could publish changes on your staging website to an Azure blob storage set up for test purposes.



Last published: 11/6/2015 12:09

Figure 10: Publish to Test feature

Please see the “Azure Publisher User Guide” for more information on this feature.

4.2 Configuring publishing profiles

Setting up publishing profiles is part of Azure Publisher configuration.

Publishing profiles define:

- Where the website data is published to in the blob storage (the container)
- What website files (all or only some) are published to the blob storage
- How the source website is published (transformation rules)
- Whether the website data uploaded to the blob storage will be downloadable

Each publishing profile must have a unique name and a unique blob container specified.

There two types of publishing profiles:

- **Primary**: A public deployment website will read from the blob container associated with this profile
- **Backup**: Additional publishing profiles, each with its won blob containers for various uses.

Azure Publisher comes with these predefined publishing profiles:

- "CD (Live)" for the primary one, set to upload to the blob container named “data”
- "All (Backup)" for the entire website backup, set to the container named “all-backup”
- "Content (Backup)" for the content-only data backup, set to “content-backup”

```
<PublishProfiles>
  <Primary name="CD (Live)" blobContainerName="data" downloadable="false">
    <SourceFiles>
      <Folder path="~/" />
    </SourceFiles>
    <FileTransforms>
      <File
skip="~/App_Data/Composite/Configuration/Composite.Azure.Publisher.config"
/>
    </FileTransforms>
  </Primary>
  <Backup name="All (Backup)" blobContainerName="all-backup">
    <SourceFiles>
      <Folder path="~/" />
    </SourceFiles>
  </Backup>
  <Backup name="Content (Backup)" blobContainerName="content-backup">
    <SourceFiles>
      <Folder path="~/App_Data/Composite/DataStores" />
      <Folder path="~/App_Data/Media" />
      <Folder
path="~/App_Data/Composite/ApplicationState/SerializedWorkflows" />
    </SourceFiles>
  </Backup>
</PublishProfiles>
```

Listing 1: Default profiles

If the named containers do not exist in the blob storage, they will be created automatically on the first publication.

If the default profiles work for you, you can skip the rest of this guide and go on to use Azure Publisher. (Please see “ Azure Publisher User Guide” for more information.)

4.2.1 Creating publishing profiles

To create a primary publishing profile:

1. From the "Content" perspective, right-click "Azure Publisher" and select "Configure".
2. On the "Profiles" tab, below <PublishProfiles>:
    a. add a <Primary></Primary> element
    b. set its **name** attribute to the name of the profile you like, for example, "CD (Live)".
    c. set its **blobContainerName** attribute to the name of a blob container where the website files are published to.
3. Within the <Primary> element:
    a. add a <SourceFiles></SourceFiles> element
    b. within the <SourceFiles> element, add one or more <Folder/> elements
    c. set their **path** attribute to the website folder path to include in publishing, for example, "~/".
4. Click "Save".

Please note that only one primary publishing profile can be defined. Normally for the primary profile, you should have one <Folder/> element set to "~/" (meaning 'all the files').

You can also have none, one or more backup publishing profiles. The steps for creating a backup publishing profile are almost the same as those for the primary profile, except that you should use the <Backup></Backup> element instead of <Primary></Primary> in *Step 2a* above.

When the user clicks the "Publish", "Republish" or “Test to Publish” button, Azure Publisher runs through all the defined profiles and publish the website data accordingly to respective blob containers in the storage.

Please see further below for information about the **<FileTransforms>** element and the **downloadable** attribute used in the example above.

4.2.2 Transforming files when publishing

Instead of copying all files ‘as is’ to the blob storage, you can configure Azure Publisher to:

- exclude specific files from being copied to the storage (skip) and / or
- replace a file when copying with the one on the target environment (skip + use)

You define file transformation rules for each publishing profile individually.

1. From the "Content" perspective, right-click "Azure Publisher" and select "Configure".
2. On the "Profiles" tab, locate a profile you want to set file transformation rules for.
3. Below the <Primary> or <Backup> element, add a <FileTransforms></FileTransforms> element.
4. Within the <FileTransforms> element, add one or more <File/> elements.
5. Set their **skip** attribute to the path to the file to skip.
6. If you want to replace a file with the one on the target environment, in addition to **skip**, set the **use** attribute to the path to the file on the target environment to use instead.
7. Click "Save".

```
<FileTransforms>
    <File skip="~/file-to-skip.txt" />
    <File skip="~/example.config" use="~/example.LIVE.config" />
</FileTransforms>
```

Listing 2: Example of transformation rules

By default, the primary publishing profile, with which Azure Publisher comes, is set to skip the Composite.Azure.Publisher.config file.

```
<FileTransforms>
  <File
skip="~/App_Data/Composite/Configuration/Composite.Azure.Publisher.config"
/>
</FileTransforms>
```

Listing 3: Example of transformation rules

4.2.3 Setting up downloadable profiles

The user can have Azure Publisher download the website files from the blob container associated with a specific profile and replace the corresponding files on the staging website.

For this you need to define a publishing profile as ”downloadable” or ”non-downloadable”.

1. From the "Content" perspective, right-click "Azure Publisher" and select "Configure".
2. On the "Profiles" tab, locate a profile you want to make downloadable or non-downloadable.
3. Add the **downloadable** attribute to the <Primary> or <Backup> element.
4. Set its value to "*true*" or "*false*" accordingly.
5. Click "Save".

```
<PublishProfiles>
  <Primary name="CD (Live)" blobContainerName="data" downloadable="false">
    <!-- the profile's settings -->
  </Primary>
  <Backup name="All (Backup)" blobContainerName="all-backup"
downloadable="true">
    <!-- the profile's settings -->
  </Backup>
</PublishProfiles>
```

Listing 4: The primary profile defined as downloadable and the backup – as not.

Please note that by default the primary profile is not downloadable, and the backup profiles are downloadable.

The downloadable profile will appear on the drop-down "Download" button. If no profile is set as downloadable, the "Download" button will be hidden.

Please note that the user must have the "Administrate" permission to see the "Download" button in Azure Publisher.