TaskStore design doc: Moving to a new Azure account  
Author: Omri Gazitt  
Created: 10/16/2011  
Last Modified: 10/19/2011

Steps to port from one Azure account to another:

* Create new Azure account/subscription using a new Live ID
* Create a new Hosted Service and a new Storage Account
* Create a new Management Certificate – first in the Azure Portal, then complete it using the Visual Studio Azure tools: use the Publish right-click action on the TaskStoreAzureService project; create new credentials; copy the certificate path to the clipboard; paste in the Azure Portal.
* Deploy the project into the new Azure account
* To enable remote desktop, create a new certificate under the Hosted Service (on my machine – search for MBARD – which is the exported certificate).
* Create a new SQL Azure database

Files that need to change when moving from one Azure account to another:

* TaskStoreAzureService\ServiceConfiguration.cscfg – the diagnostics provider connection string has the storage account and key (one for each role)
* Startup\DeployBlob.exe.config – the config has the storage account and key (this is for the Install.cmd script to be able to download large installation files from azure storage and run them as part of the deploy process)
* TaskStoreWebMvc3\web.config – the SQL Azure connection string is embedded in this file (under the key “TaskStore”)
* TaskStoreMailWorker\app.config – the SQL Azure connection string is embedded in this file (under the key “TaskStore”)

Before these files are updated as part of a re-deploy, it’s important to copy data from the old SQL Azure and Azure Storage accounts into the new ones:

* In SSMS, make sure that the new database has a login and a user corresponding to the login and user for the old database. If not, use the CREATE LOGIN and CREATE USER scripts to accomplish this on the new database.
* In SSMS, issue the following command to the new database: CREATE DATABASE TaskStore AS COPY OF <oldsrvname>.TaskStore
* Mstsc into the newly deployed Azure TaskStoreWebMvc3 instance, and issue the following commands:
  + E:
  + mkdir \files
  + cd E:\sitesroot\0\startup
  + deployblob /downloadFrom tsinstall /downloadTo \files
  + notepad deployblob.exe.config // change the account creds to the new storage account
  + DeployBlob.exe /uploadTo tsinstall /uploadFrom e:\files\\*

Now we are ready to do another deploy which will completely switch over to the new SQL Azure and Azure storage accounts.