JSON Parsing in USQL:

In USQL there are built-in extractors for parsing text, comma delimited or tab delimined files. Once again, parsing JSON becomes problematic.

There is a solution built into USQL, write some C# code to extend it or use someone else's C# code to extend USQL. Since we wanted to parse JSON, fortunately there are libraries available on github containing the information required to do it. Download the github package and open up the Microsoft.Analytics.Samples project in Visual Studio. When I did this the first time, there was a problem loading the Newtonsoft.Json reference, so I right clicked on the references and downloaded the missing parts again. Build the solution and check out the code in the directory …Examples\DataFormats\Microsoft.Analytics.Samples.Formats\bin\Debug\ . There will be two DLLs, Microsoft.Analytics.Samples.Formats.dll and Newtonsoft.Json.dll. These dlls then need to be registered in Data Lake Analytics and locally if you chose to run your USQL locally. As at some point the goal is to run from within Data Lake analytics, you will need to copy both of these dlls to the data lake. I created a folder for the dlls called Assemblies, and ran this command

```
USE DATABASE [master];
CREATE ASSEMBLY [Newtonsoft.Json] FROM
@"/Assemblies/Newtonsoft.Json.dll";
CREATE ASSEMBLY [Microsoft.Analytics.Samples.Formats] FROM
@"Assemblies/Microsoft.Analytics.Samples.Formats.dll";
```

Notice I told the USQL where to find the dlls, in the Assemblies folder. This step only needs to be completed once per data lake. After this job successfully runs, then the dlls which allow the JSON to be parsed, can be referenced.

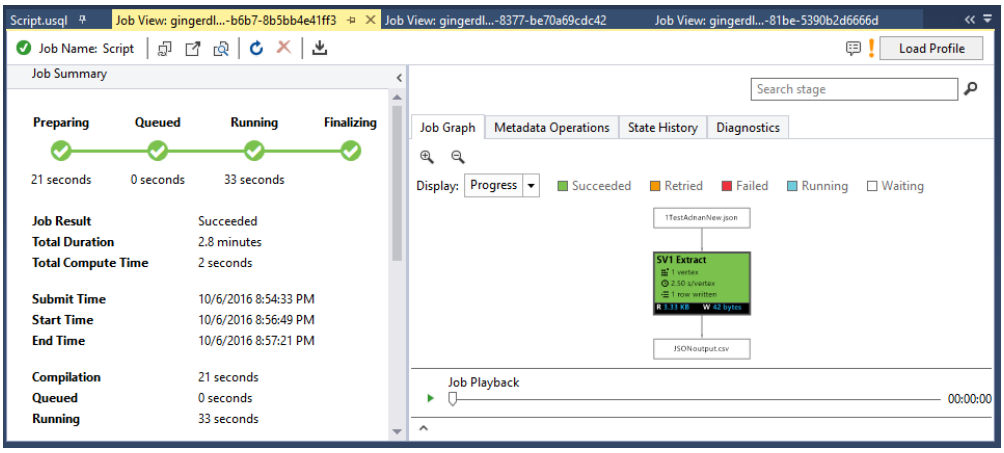
Here is my sample JSON, which I have copied to the folder Samples/Data/TestNew.Json, in the Data Lake

```
{
"appInstanceId": "357ced1e-cf05-459c-9317-794bq24f61c2",
"firmwareVersion": "1.0.2.4",
"serialNumber": "254542-694967",
"Side": "0",
"Latitude": "33.8848744",
"Longitude": "-128.403276",
"GeneratedDate": "2016-10-04T21:18:19Z"
}
```

Now that I have added the JSON to the Data Lake and the assemblies have been added, I can write some USQL to Parse the JSON. First I will need to reference the libraries, then create a schema, as there is no schema for a Data Lake. After those steps are completed, it's possible to write SQL to query a JSON file. There is no UI to look at the results, so the results will be writing to a file. I am going to output the data to a csv file called JSONOutput.csv. Here's the code to do that.

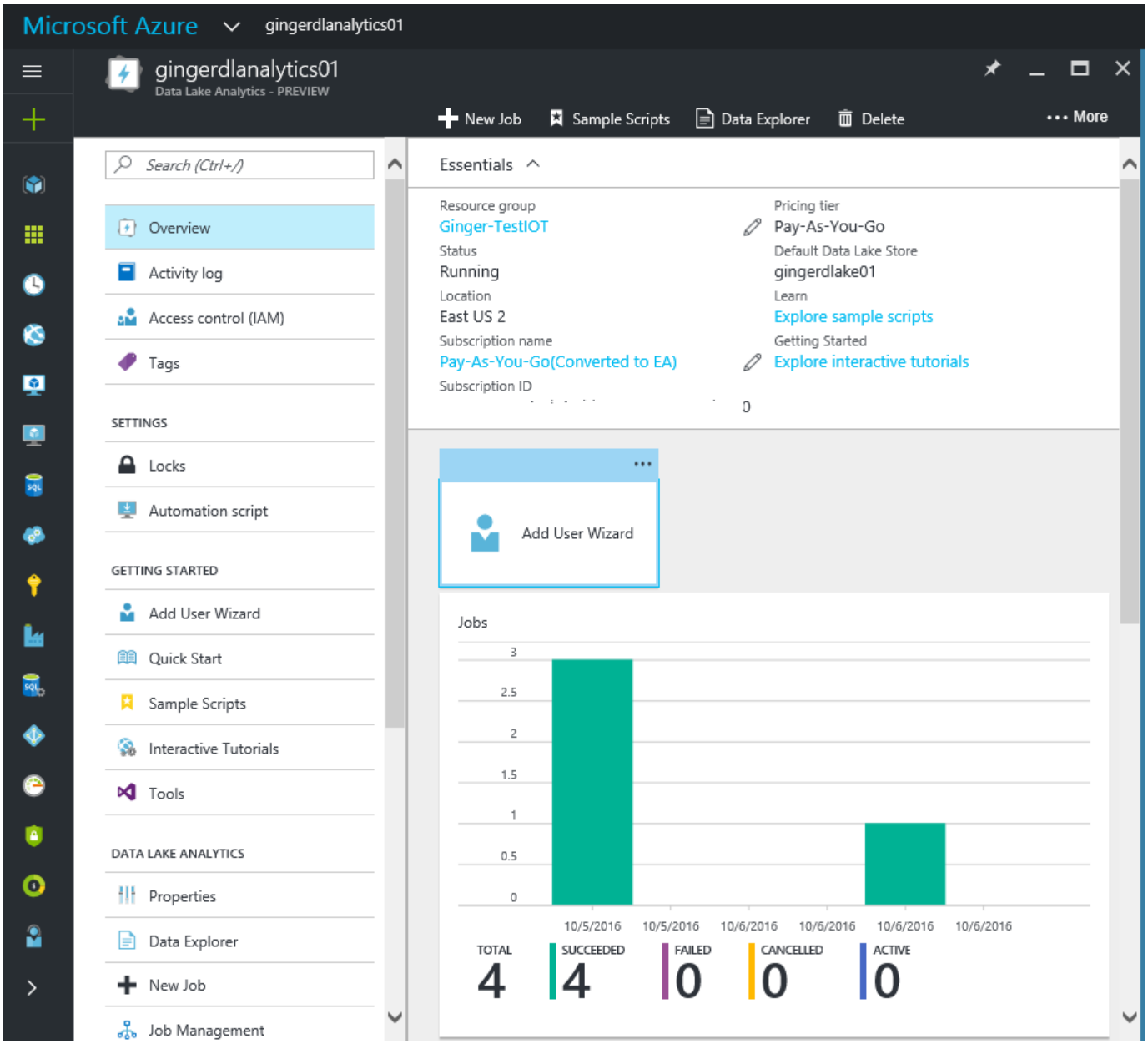
```
REFERENCE ASSEMBLY [Newtonsoft.Json];
REFERENCE ASSEMBLY [Microsoft.Analytics.Samples.Formats];

DECLARE @infile string="/Samples/Data/TestNew.json";

@logSchema =
EXTRACT name string
, appInstanceId string
, firmwareVersion string
, serialNumber string
, Side string
, Latitude float
```

```
, Longitude float
FROM @infile
USING new Microsoft.Analytics.Samples.Formats.Json.JsonExtractor();

@testthis = SELECT appInstanceId
, COUNT(*) AS LocationCount
FROM @logSchema
GROUP BY appInstanceId;

OUTPUT @testthis
TO "/Samples/Data/JSONoutput.csv"
USING Outputters.Csv();
```



Using Visual Studio, I am running the USQL Job. There isn't much data to parse, and you can see in the summary widows that it took 21 seconds to prepare, and 33 seconds to run.

When go to the web and look at the Data Lake Analytics page, I can also see that the job completed. I have noticed that this appears pretty close to the same time on the web and on visual studio.



Clicking on the bar graph represented by today will allow me to select the job which ran, showing the same screen as appears in Visual Studio.