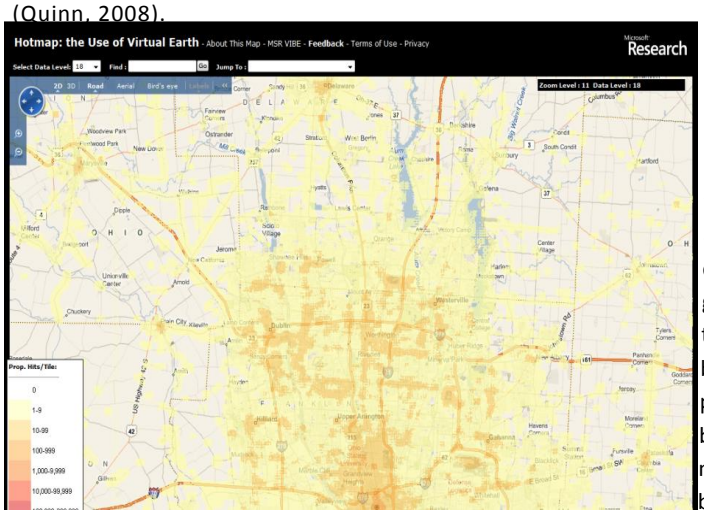
# ArcGIS Server Cached Map Tile Tracking

## Introduction

ArcGIS Server enables organizations to create cached map services, which consist of pre-rendered tiles that enable developers and administrators the ability to create scalable, high performance web applications. A considerable amount of planning and preparation goes into prioritizing layers, planning the scale levels, coordinate system, and cartography before burning the cache. When administrators and map cache authors develop map caches, it may not be practical or efficient to cache all data for the organizations entire study area. Typically large caches, those taking 8 hours or more to build should be broken up into smaller jobs. Additionally, organizations should develop large caches to "anticipate" high traffic areas while excluding low traffic that may include rural or uninhabited areas, like forests, deserts, large water bodies, and even remote farmland. This strategy of selective caching can "save time," "disk space," and the areas excluded from the initial cache can be cached "on-demand" (Quinn, 2008).




When caching is complete and applications are consuming cached map services, questions remain for Web Administrators to answer. Who is using the cache? Where are people going when they visit a map that uses map tiles? It would be nice to know what tiles people are viewing and even better to model the parts of a map cache that have already been viewed. This idea, known as descriptive modeling shows where users have already viewed and it is useful for deriving and validating predictive models (Quinn, 2008). Microsoft Research has done work in this area to show the use of Virtual Earth tiles through a heat map view. Other use cases for descriptive analysis include economic development. If a list of properties zoned commercial, industrial or mercantile are available for purchase but are not being visited by perspective business owners, economic development organizations can look at tile traffic to see if people are even looking at less desirable properties. They could then use the information gathered to develop or propose incentives to encourage perspective businesses to develop in areas that may be deemed less desirable.

The purpose of this document is to describe how to implement Descriptive Tile Analysis Modeling so that organizations can track tile usage using ArcGIS Server and Geoprocessing. Here is a link to a site showing tile usage, developed using the instructions and workflow found in this document. The workflow is actively tracking all tile requests that include zoom in, zoom out and pan clicks found in this link.

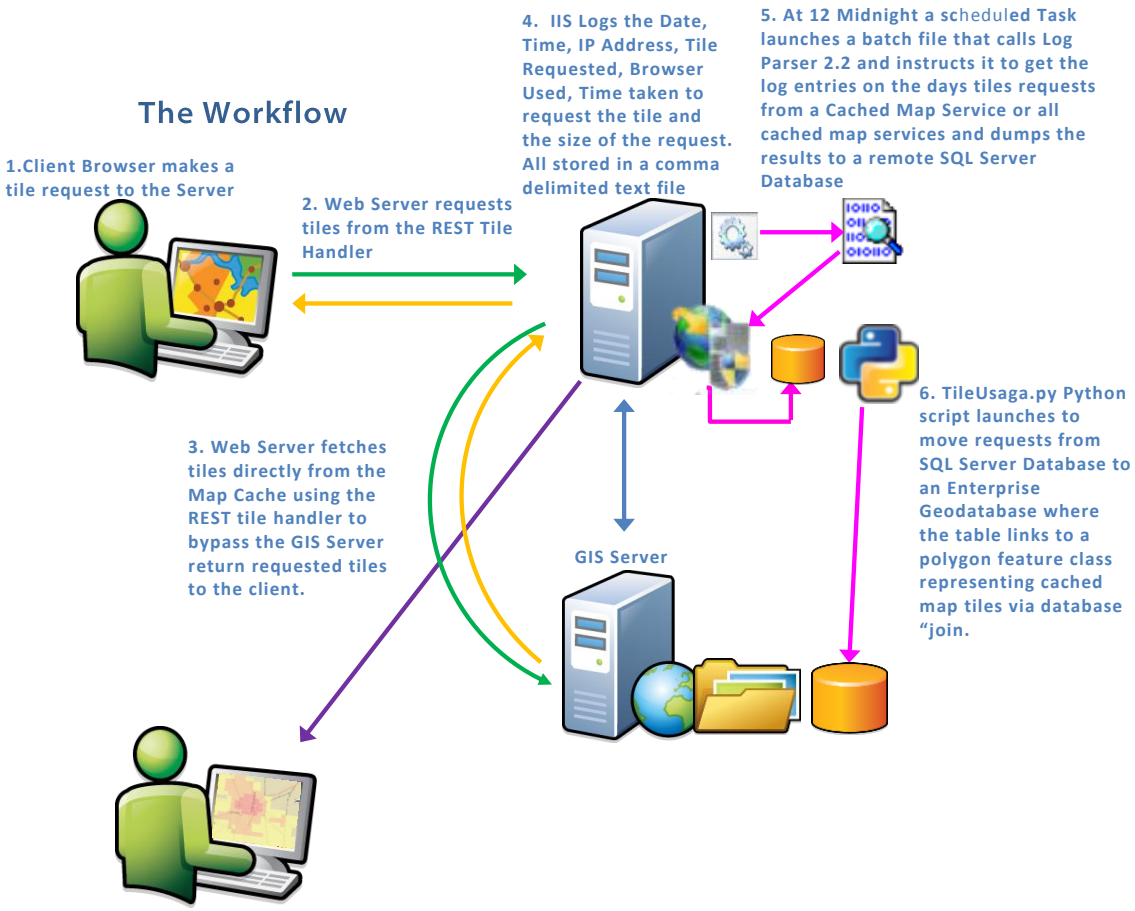
## Requirements

The following is a list of software and tools used to create the tile tracking workflow used by ArcGIS Server. This document outlines how to track tile usage using ArcGIS for Server and Microsoft Windows Server.

1. Administrative Privileges on the Web Server, ArcGIS Server
2. DBO privileges to the database.
3. ArcGIS Server 10.0, 10.1 and ArcGIS for Server 10.2 Basic, Standard and Advanced
    a. Operating System: See the ArcGIS for Server supported platforms (Windows)
    b. Supported Application/web servers Web Adaptor for IIS
    c. Batch File and DOS
    d. Task Scheduler 1.0
4. ArcGIS for Desktop Advanced 10.2
5. ArcPy Python Module
6. Microsoft SQL Server 2008 R2 or Microsoft SQL Express 2008 R2
    a. SQL Server Native Client
7. ArcSDE Enterprise, Workgroup or Personal Edition
8. Microsoft Log Parser 2.2

The key component to implementing tile tracking is Log Parser 2.2 Database Logging. It has the ability to query text, csv and database files using SQL queries. This allows us to parse the IIS logs, and easily identify map service tile requests. Query results can be output to charts, reports, CSV or SQL Server. In the case of SQL Server, the first time a query is executed a table can be created and each successive time that it runs, records can be appended to the table. Finally, Log Parser SQL statements are executed from command line or batch file, which in turn can launched from Windows

## The Workflow




# Tile Tracking Setup Instructions

## Step 1

Unzip the TileTracking.zip file to a location on your server.

## Step 2

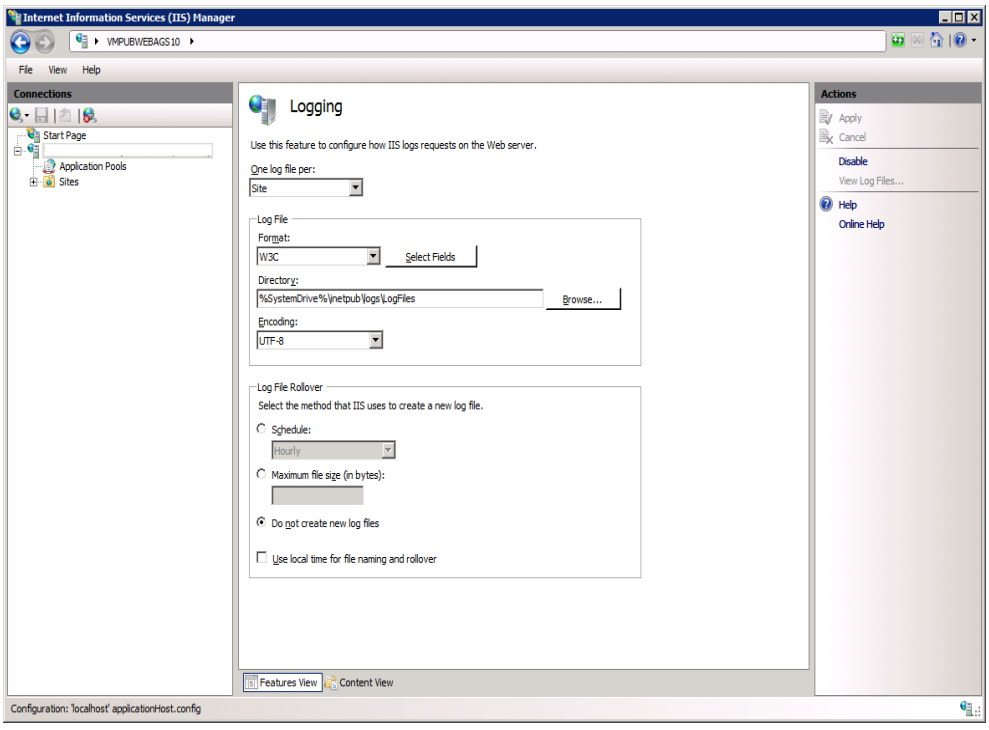
Download and install Log Parser 2.2 on the Web Server/Web Adaptor machine (IIS).

## Step 3

By default, IIS creates a new log file for everyday of the week. Depending on how long IIS has been running and how often IIS is set to clean up or remove log files this means that the log directory could contain hundreds of files. This also means the batch file executing Log Parser has to account for a new log file name every day. To avoid this, configure IIS use a single log file. Please consult your server administrator before making this change to the web server. If this is a problem, rewrite the batch file to utilize a variable to catch the day of the week and know what log file to look for.

To modify IIS to create a single file you will need to have administrative privileges

Go to Start| Administrative Tools | Internet Information Services (IIS) Manager | Click Continue on User Account Control Dialog *Administrative Privileges Required* |Select the Web Server| Choose Logging | Do Not Create New Log Files |Apply | Close IIS.



## Step 4

Create the IISLogsLP Database on the server where SQL Server resides. Browse to the folder .\TileTracking\SQL\ and open the CreateIISLogsLP_DB.sql with a text editor like notepad.

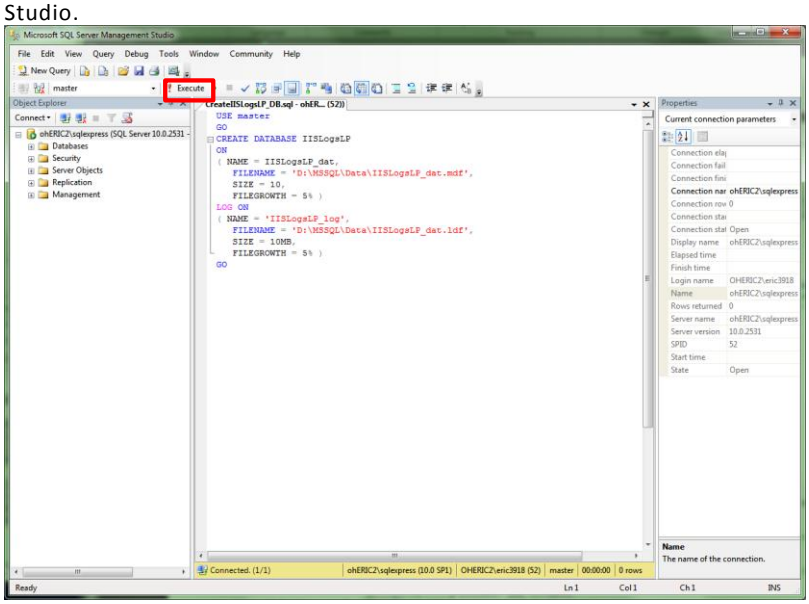
```
USE master
GO
CREATE DATABASE IISLogsLP
ON
( NAME = IISLogsLP_dat,
    FILENAME = 'D:\MSSQL\Data\IISLogsLP_dat.mdf',
    SIZE = 10,
    FILEGROWTH = 5% )
LOG ON
( NAME = 'IISLogsLP_log',
    FILENAME = 'D:\MSSQL\Data\IISLogsLP_dat.ldf',
    SIZE = 10MB,
    FILEGROWTH = 5% )
GO
```

**Comment [EJR1]:** Modify the directory to point to the location the SQL Server Data file will live.

**Comment [EJR2]:** Modify the directory to point to the location the SQL Server Log File will live.

Save the Text file. Double click on the text file to launch SQL Server Management Studio. Management Studio will open with the newly edited SQL code loaded and ready to execute.

Click the Execute button to create the IISLogsLP Database and close SQL Server Management Studio.



## Step 5

The script below will create the IISLogsLP table and it will import the fields and values of dateTime, IP Address, Time Taken to generate a request, and the URL of the request (Map Cache Tiles) from the IIS Log file u_extend1.log. Additional fields can be imported for more information on fields to be logged by IIS 7 click here. You cannot create additional fields in the IISLogsLP database outside of the fields that are contained in the IIS Log files. If you try to include other fields, Log Parser 2.2, will error. Modification of the IISLogsLP table can take place once the geodatabase conversion takes place in step 10. To get started, you will be selecting records that match the tile service you wish to cache since its inception. The final scheduled task that runs daily will select requests made in the previous 24 hours for the chosen map cache.

To ensure that Log Parser has records to work with, go to the ArcGIS Services Directory usually located at http://your domain or server name/arcgis/rest and access the cached map service you wish to track. View the service in ArcGIS JavaScript or ArcGIS.com and zoom in, out or pan the service to generate tile requests.

To create the IISLogsLP table with Log Parser 2.2. Modify the blue highlighted lines using the comments on the right hand margin. When copying the commands do not copy the yellow highlighted text that indicates the number of commands to execute. Line 2 is one long line of code that Microsoft Word is wrapping:

```
Line 1: cd /D "C:\Program Files\Log Parser 2.2"

Line 2: LogParser.exe "SELECT TO_TIMESTAMP(date, time) AS dateTime, c-ip,
time-taken, cs-uri-stem FROM c:\inetpub\logs\LogFiles\w3svc1\u_extend1.log TO
IISLogsLP WHERE cs-uri-stem LIKE
'/ArcGIS/rest/services/Delaware/DelCoBaseMap_Exploded/MapServer/tile%'"
-o:SQL -oConnString: "Driver={SQL Server Native Client 10.0};
Server=.\sqlexpress; Database=IISLogsLP;Trusted_Connection=yes;" -
ignoreMinwarns:OFF -createTable:ON -maxStrFieldLen:8000
```

**Comment [EJR3]:** URL of the map service we want to track. I am tracking a map service called DelCoBaseMap_exploded which is located in the folder Delaware.

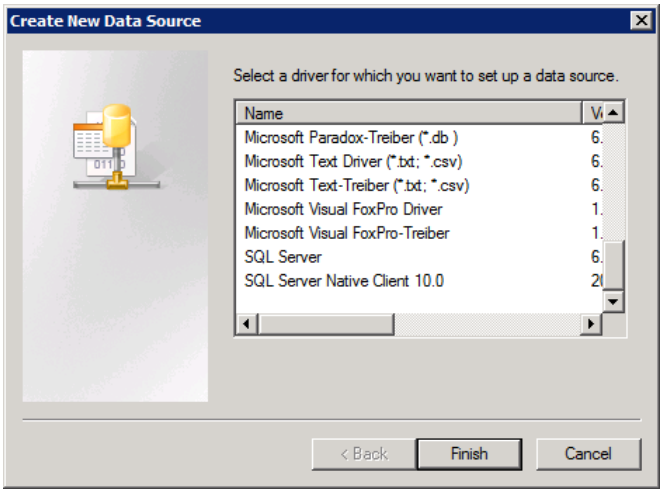
**Comment [EJR4]:** SQL Server Native Client Driver to use. See the note on how to choose the correct driver

**Comment [EJR5]:** .\sqlexpress indicates that I am logging into a SQL Express database and the "." Indicates that the server is the localhost server, which is also the webserver.

**Special Note**

The version of SQL server you have determines the SQL Native Client Drivers you can call. The easiest way to determine this information: Start | Administrative Tools | Data Sources (ODBC Drivers) | Click the Continue button on the User Account Control Dialog *this requires Administrative Privileges* | Click on the System DSN tab | Click Add | Locate the SQL Native Driver and use the full name as you see it in your dialog |

Click Cancel to close the dialog




## Step 6

Open a command prompt.

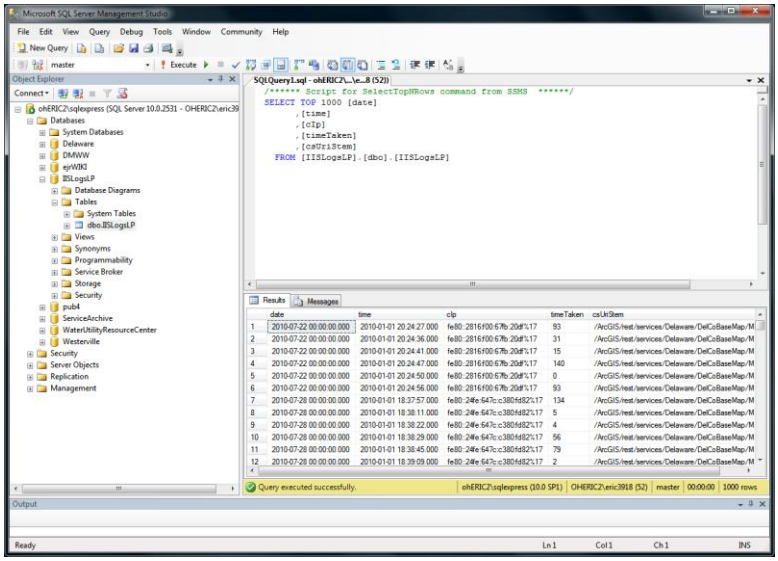
Start | Run| Type “CMD” in the command prompt | Enter

Copy the commands we modified in Step 5 and paste them into the command prompt. To paste lines into a command prompt:

Click on the title bar | right click | Edit | paste | hit enter to execute commands

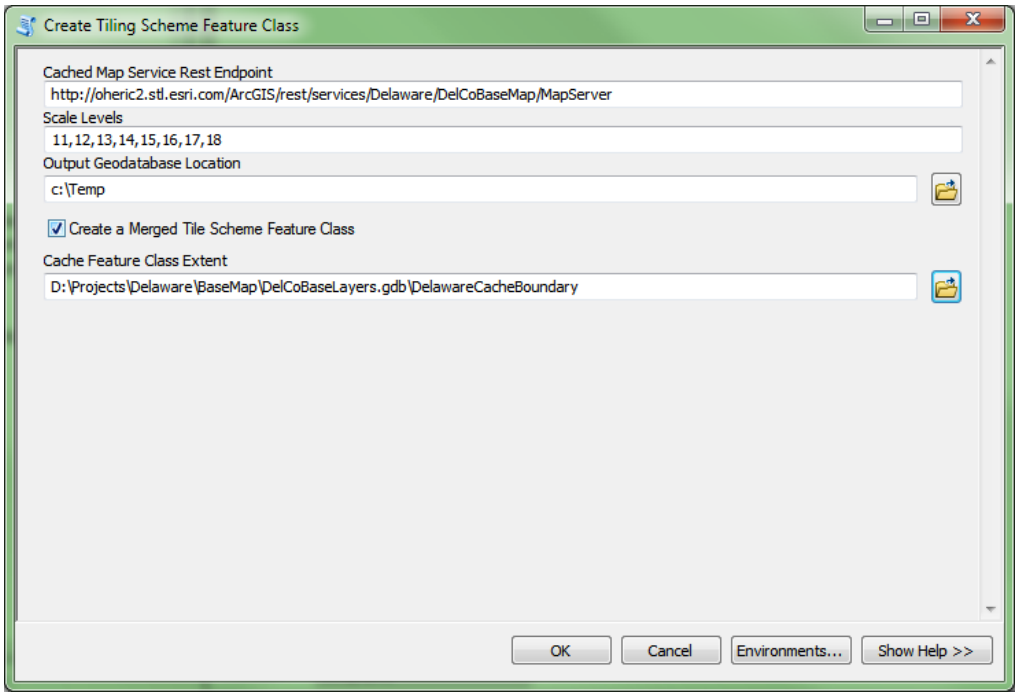
Open SQL Server Manager

Start | All Programs | Microsoft SQL Server 2008 |SQL Server Management Studio| Connect to the Database Server |Expand the Databases Node | you should see a database called IISLogsLP.



## Step 7

Install Create Tiling Scheme Feature Class python toolbox included with this package and run it to create the Tiling Scheme geodatabase. This script will create a feature class representing of each scale level in the cache. Additionally, the script creates a composite feature class containing features from every scale level. Each feature class contains the polygons representing the map cache and fields to track and the Scale Level, Row ID, Column ID and Tile ID fields.




Make sure to check the "Create a Merged Tile Scheme Feature Class" option

## Step 8

Create an SDE Geodatabase and load the Merged Tile Scheme Feature Class to facilitate tile tracking.

1. Open ArcCatalog and either create a new empty SDE Geodatabase or use an existing SDE Geodatabase.
2. Locate the TileScheme File geodatabase created in step 7. Expand the node to list all of the feature classes. Locate the CacheMapTilingScheme feature class. Rename the feature class to match the name of the service that you are tracking.
3. Right click on renamed CacheMaptilingScheme feature class and choose copy.

4. Right click on the Tile Tracking Enterprise database in the Database Servers Node, choose Paste, and click OK to confirm the data transfer.
5. Make sure that the ArcGIS Server account has access to this geodatabase. The ArcGISSOC account needs to have a minimum of "Read Only" privileges. If privileges are not granted to this database then ArcGIS Server will not be able to display the heat map.
6. Copy and paste the SDE connection file located in C:\Users\%username%\AppData\Roaming\ESRI\Desktop10.0\ArcCatalog to the Data Connections folder in the Tile Tracking workspace. If you cannot find the file, make sure Windows is set to show hidden files and folders

## Step 9

Create the database connection to IISLogsLP in ArcCatalog for the TilingUsage.py Python script. Open ArcCatalog > Open the Database Connections Node in the Catalog Tree > Add OLE DB Connection > SQL Server Native Client 10 > Next >

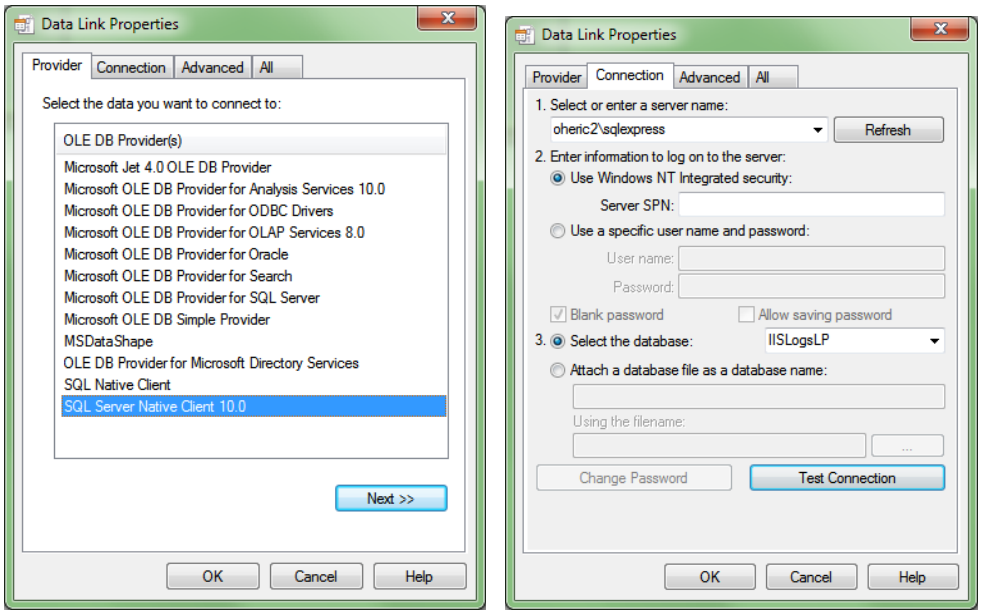
1. Enter a Server Name: hostname\Database Instance

2. Enter information to log on to the server: Use Windows NT Integrated Security

3. Select a Database: IISLogsLP

Click the Test Connection Button.

If the connection test succeeds then click ok to close the dialog otherwise troubleshoot the problem.



Name the connection in ArcCatalog to SQL@NativeConnection@IISLogsLP.odc. Move the connection file into the "DataConnections" folder found in the "TileTracking" directory. To copy the file go to C:\Users\%username%\AppData\Roaming\ESRI\Desktop10.0\ArcCatalog and locate the SQL@NativeConnection@IISLogsLP.odc connection string and copy and paste the connection to the Data Connections folder. If you cannot find the file, make sure Windows is set to show hidden files and folders

## Step 10

Configure the Tile Usage Python script. Go to %\TileTracking\Python\TileUsage\ and open the TileUsage.py using notepad or a text editor.

Using "Find and Replace" in Notepad, search for the line beginning with dbo_IISLogsLP. It should be on line 49 and will look like:

```
#******************* Edit the locations of the SQL Server Database Connection
and Tile Scheme Geodatabase**************************
dbo_IISLogsLP =
"D:/Projects/Caching/TileTracking/DataConnections/SQL@NativeConnection@IISLog
sLP.odc/dbo.IISLogsLP"
TilingScheme_gdb =
D:/Projects/ags_Server/Caching/TileTracking/DataConnections/DC@oheric2@TileTr
acking.sde"
tileHits = 'TileTracking.DBO.IIS_TileHits'
Frequency = 'TileTracking.DBO.TileHit_Frequency'
#Do Not Edit Below This Comment
```

**Comment [EJR6]:** Location of the SQL@NativeConnection@IISLogsLP.odc connection file

**Comment [EJR7]:** Location of the SDE connection file to connect to the TileTracking SDE Geodatabase.

**Comment [EJR8]:** Fully qualified name for the TileHits Table

**Comment [EJR9]:** Fully qualified name for the Tile Hit Frequency Table

Update the path to dbo_IISLogsLP to reflect the location of the SQL@NativeConnection@IISLogsLP.odc set up in Step 9.

Update the next line, which is the location of the SDE connection file saved in step 8.

Provide the fully qualified table names for the Tile Hits and Tile Hit frequency tables, created by the "TileUsage.py" python script.

Fully qualified names follow the pattern of Database.User.TableName. The user should be DBO, as you are the owner of the TileTracking SDE Geodatabase. If you are not sure, you can open ArcCatalog, go to your DataConnections folder, and connect to the SDE connection file you created in Step 9. Look at the name of the CacheMapTilingScheme Feature Class. The Database name and User DBO should precede it. If the user is different then that is the name you should use for the user.

## Step 11

Test the TileUsage.py to create the IIS_TileHits and TileHit_Frequency tables in the DC@HostName@TileTracking.SDE geodatabase.

Open a command prompt.

Start | Run| Type “CMD” in the command prompt | Enter

Copy the following command line and paste into the command prompt. To paste lines into a command prompt

Click on the title bar | right click | Edit | paste | hit enter to execute the lines

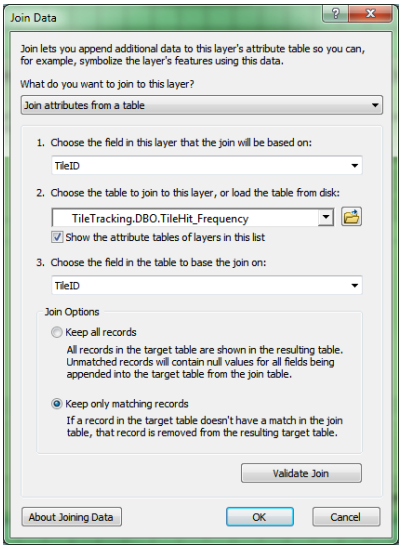
```
C:\Python27\ArcGIS10.2\python.exe "%Your Path%\TileTracking\Python\TileUsage\TileUsage.py"
```

Open ArcCatalog and locate the Database Connections Folder. Open the SDE connection file and view the contents of the database. You should see two tables named IIS_TileHits and TileHit_Frequency. The IIS_TileHits table imports the data collected by Log Parser 2.2 stored in the IISLogsLP table. TIleHit_Frequency is the result of Geoprocessing Frequency analysis to determine the frequency of tile viewings.

## Step 12

Browse to %\TileTracking\CacheUsageHeatMap\ and open the CacheUsageHeatMap.mxd. Add the CachedMapTilingScheme Feature class from SDE. Do not worry about setting the symbology of this feature class. The ArcGIS Server for JavaScript API application included with this sample will dynamically assign symbology to the data in this map on the fly.

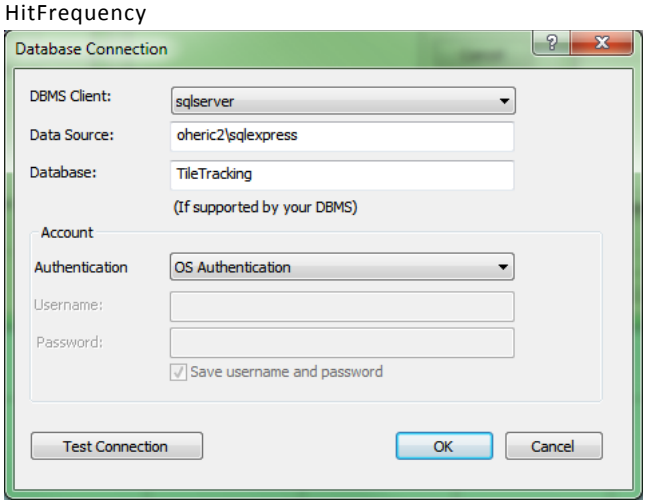
1. Right click on the feature class and choose Properties.
2. In the properties Dialog choose Joins & Relates > Joins > Add > Use the following graphic below as a guide to set the join:



3. Click OK and Ok on the Properties Dialog

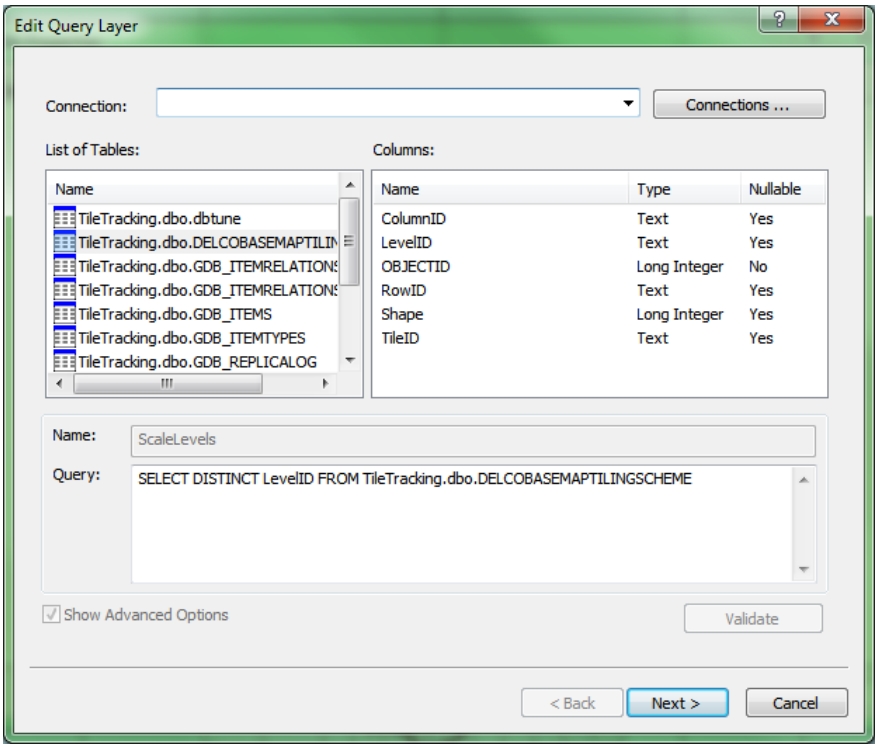
Notice the Join Options are set to use Keep only Matching Records. This is critical to the overall performance of the web application used to monitor which tile usage. The feature class used to model the cache for Delaware County, Ohio contains over 500,000 tiles and by "Keeping only matching records" lowers that number significantly. As new tiles are viewed they will be accounted for and added every night when the TileUsage.py and Log Parser are run at their scheduled time.

4. Add a Query Layer to the map.
    a. The query layer will provide the web application with the list of scales that it needs in order to allow the end user to pick the scale they wish to display tile usage.
    b. Go to File > Add Data > Add Query Layer
    c. Click on the Connections Button and Choose New Connection
    d. Select the DBMS Client. I used SQLServer
    e. Data Source: If you are using an enterprise SDE the datasource will be the Hostname. If you are using Workgroup SDE the data source will be Host\Instance.
    f. Type in the name of the database which is TileTracking
    g. In most cases, the connection type is always OS authentication.
    h. Click Test Connection to ensure that you can connect.
    i. Click OK once you successfully connect and name the connection HitFrequency




    j. Choose the HitFrequency connection from the dropdown menu. Once selected you should see all of the tables that are part of the TileTracking Database.
    k. Set the Name of the Query Layer as ScaleLevels

l. Locate the TileTracking.dbo.DelCoBaseMapTilingScheme table
m. In the query box type the following SQL Query: SELECT DISTINCT LevelID FROM TileTracking.dbo.DelCoBaseMapTilingScheme
n. Click next and check the LevelID check box. The query layer will create Unique Identifier field based on rows returned from the query layer.
o. Click Finish

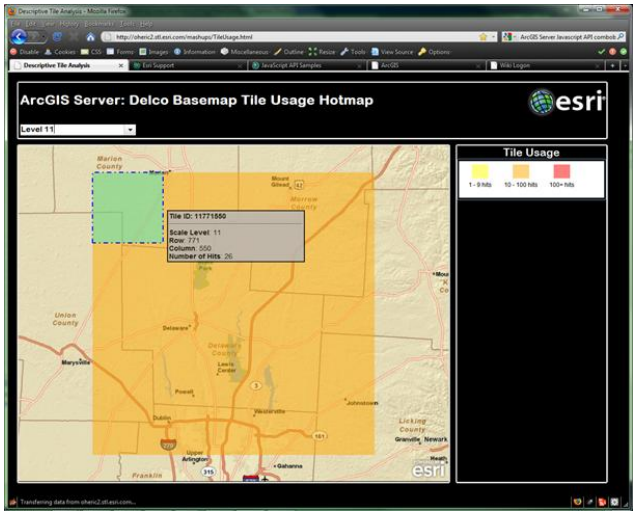



5. Save the Map as CacheUsageHeatMap
6. Go to File > Share As > Service to publish map as a Map Service.

## Step 13

This workflow provides a custom application mashup that combines the World Streetmap from ArcGIS Online with the service published in step 12 using the ArcGIS API for JavaScript. The application comes with a configuration file to make setup simple and customizable. The application is uses a Class Breaks Renderer to symbolize tiles on the fly based on the number of times each tile has been viewed. Additionally, the sample makes use of a Legend Widget, Feature Layers, Info Windows and an excellent example by Kelly Hutchins of ESRI that allows the application to populate a combo box with unique values (scale levels) retrieved using a query layer set up in step 12.

1. On the Web Server create a directory where the web application will reside, for this sample, the directory “Mashups” on c:\inetpub\wwwroot\Mashups was used. This directory can live anywhere and its name can be whatever you want. When IIS registers it as a virtual directory, ArcGIS Server will know what to do with it.
2. Using IIS make the Mashups directory a virtual directory.
3. In the home directory of this sample locate the inetpub folder %\TileTracking\Inetpub\wwwroot and copy the AppFiles Folder, Images Folder and TileUsage.html and paste them into the Mashups Directory or whatever you called it.
4. Open a web browser and navigate to the rest endpoint of the service created in step 12. I used http://oheric2.stl.esri.com/ArcGIS/Rest/Services/Delaware/CacheUsageheatMap/MapServer
    a. This page will provide you with much of the information you need to complete the config.js page.
5. Using a text editor like NotePad++ or Aptana Studio open the config.js file located in the AppFiles Folder that you just copied. I like either of these editors because they color code the JavaScript and make it easy to tell strings from numeric values also make comments easy to identify.
    a. Use the comments to help modify each section of the application and refer to the rest endpoint in the previous step for extent and service information.
    b. Modify each section of the configuration file as you see fit.
    c. Locate the image folder in the Mashups Directory. Place an image, such as your organization logo into the folder. The image needs to be “.PNG” and its dimensions should be 150 pixels wide by 70 pixels high. Name the image logo.png and the application will load it into the upper right hand corner of the application.



d. Once the sections are modified, open a web browser and type the url: http://<Your Server>/mashups/TileUsage.html

## Step 14

Configure the batch file and schedule the task to run

Open the Batch File make sure the directory names are correct. You may need to change some directories to reflect the location of your applications.

Browse for %\TileTracking\BatchFiles\TileTracking102.bat. Right Click the file and choose Edit.

The batch file should open in notepad. In Notepad go to View and make sure that Word Wrap is unchecked. The batch file consists of 16 lines of code, which also includes comments.

```
@ECHO ON
REM Report the date and time for logging
ECHO %DATE% %TIME%

Rem Stop Tileusage Service so that the IIS Logs can be refreshed with the
previous days activities
cd /D "C:\PYTHON27\ArcGIS10.2\"

python.exe "C:\Program Files\ArcGIS\Server\tools\admin\manageservice.py" -u
erodenberg -p mapmaker -s http://localhost:6080 -t -n CacheUsageHeatmap -o stop
Rem Stop Tileusage Service so that the IIS Logs can be refreshed with the
previous days activities

Rem Update IISLogsLP with the previous days activites
cd /D "C:\Program Files\Log Parser 2.2"


LogParser.exe "SELECT TO_TIMESTAMP(date, time) AS dateTime, c-ip, time-taken,
cs-uri-stem FROM c:\inetpub\logs\LogFiles\w3svc1\u_extend1.log TO IISLogsLP
WHERE cs-uri-stem LIKE
'/ArcGIS/rest/services/Delaware/DelCoBaseMap_Exploded/MapServer/tile%%' AND
dateTime >= SUB(SYSTEM_TIMESTAMP(), TIMESTAMP('0000-01-01 23:59:59','yyyy-mm-dd
hh:mm:ss'))"
-o:SQL -oConnString: "Driver={SQL Server Native Client 10.0};
Server=.\sqlexpress; Database=IISLogsLP;Trusted_Connection=yes;" -
ignoreMinwarns:OFF -createTable:ON -maxStrFieldLen:8000

Rem Update Tile Usage Geodatabase
C:\Python27\ArcGIS10.2\python.exe
"C:\ags_share\TileTracking\Python\TileUsage\src\TileUsage.py"

Rem Restart Tile Usage Map Service
python.exe "C:\Program Files\ArcGIS\Server\tools\admin\manageservice.py" -u
erodenberg -p mapmaker -s http://localhost:6080 -t -n CacheUsageHeatmap -o
start
```

**Comment [ER10]:** Stopping the CacheUsageHeatMap Service

**Comment [EJR11]:** Changing the directory to the Log Parser Home Directory

**Comment [EJR12]:** 2 %% signs required for DOS bat to interpret the % as a partial string wildcard. This was 2 days of pain and suffering

**Comment [EJR13]:** Selecting the Date, Time IP Address, Request Speed, and Requested Tile fields from IIS Log File as long as the URL is coming from the selected Cached Map Service

**Comment [EJR14]:** Get the last 24 hours of records.

**Comment [EJR15]:** Scheduled Task will execute using my username. I know that log parser will be able to use my SQL connection as I am a database owner

**Comment [EJR16]:** Changing directories to the python directory

**Comment [EJR17]:** Calling the Tile Usage Script

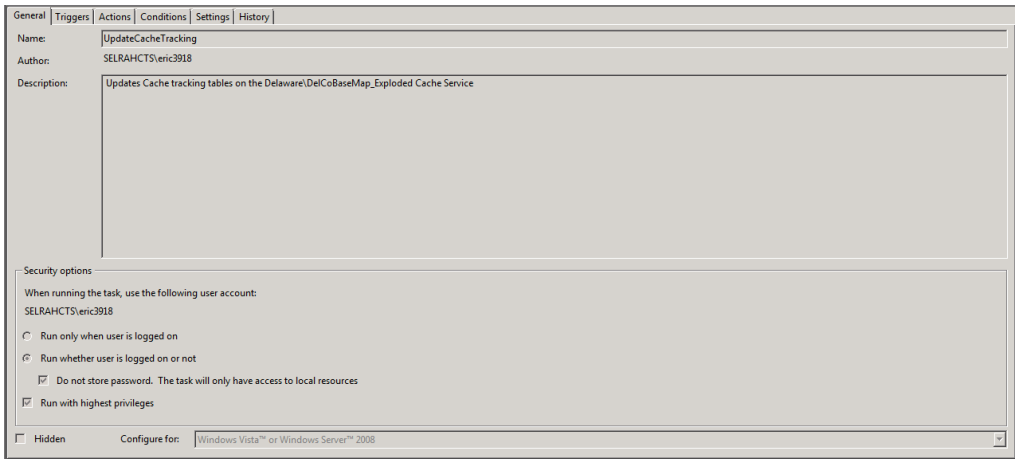
**Comment [ER18]:** Restarting the CacheUsageHeatMap Service

## Step 15

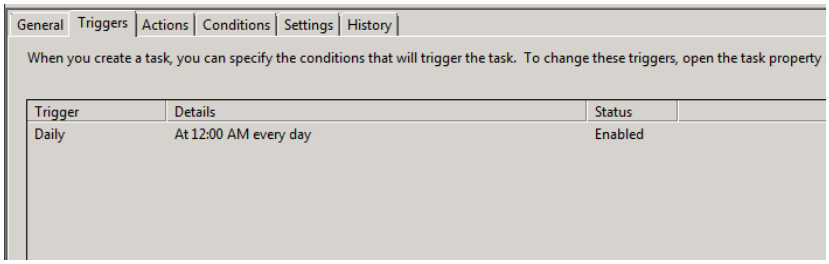
Schedule the task to run… preferably during off hours.

Start | All Programs | Accessories| System Tools |Task Scheduler |Click Continue on User Account Control Dialog *Requires Admin Privileges* |Create New Task

Name the Task and set to run whether logged in or not.  Do not store your password and use local resources.  Run all processes with the highest privileges… this will insure that Windows User Account Control does not prevent this process from running.




On the Triggers Tab set the task to run at a set time every day.

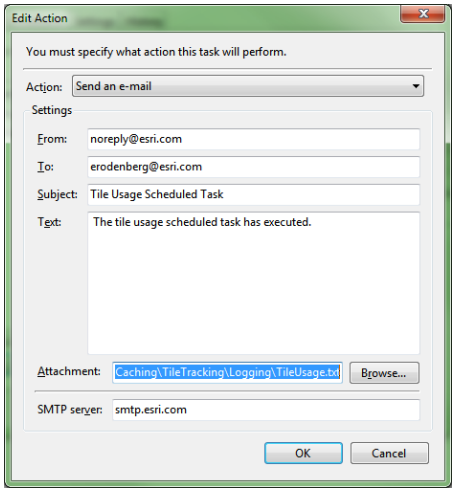



Click the Actions Tab and choose New  and for the Action choose the default “Start a program.”  On the Program/script entry click the browse button and browse to the Logging.Bat, which will launch the TileTracking.bat file configured in step 9 as well as TileTracking Log file stored in the Logging Directory.  This log file allows us to see the results of the scheduled task that we run on a nightly basis.

Set the Start In line to “.\TileTracking\BatchFiles\” and choose OK.

***Optional*** Email the log file to yourself or the ArcGIS Server Administrator.

On the actions Tab choose New > Action: Send an Email




Set the attachment to the ".\TileTracking\Logging\TileUsage.txt" and every night when the process is finished, an email report with the results of log parser and the tile usage Geoprocessing tasks will be waiting for you in your email. Contact your IT staff for your organization's SMTP server information when setting the email report up.

Click OK and close the Task Scheduler. You have successfully configured a workflow to track tile usage of a cached map service in ArcGIS Server

If you would like to see this workflow in action, click on the link to go to ArcGIS.com and visit a tracked, published cached map service. As users zoom in, out or pan around the map, IIS is recording every tile that they hit and being updated to the IISLogsLP database nightly. Additionally, the CacheUsageHeatmap Service shows the tiles "hit," to see a live demonstration click here. To view the scale levels click the details button then choose Show Contents of Map. Click on the "CacheUsageHeatmap" layer name to expose the ability to toggle on or off the individual scale levels visited. You can also use the identify tool to see the individual tiles and the number of times the tile has been viewed.

## Works Cited:

Quinn, S. (2008, September 10). *Predicting popular areas of a tiled Web map as a strategy for server-side caching.* Retrieved September 8, 2010, from Pennsilvania State University: http://www.google.com/url?sa=t&source=web&cd=2&ved=0CBkQFjAB&url=https%3A%2F%2Fwww.e-education.psu.edu%2Ffiles%2Fmgis%2Ffile%2FQuinn_Peer_Review_Presentation_20080901.ppt