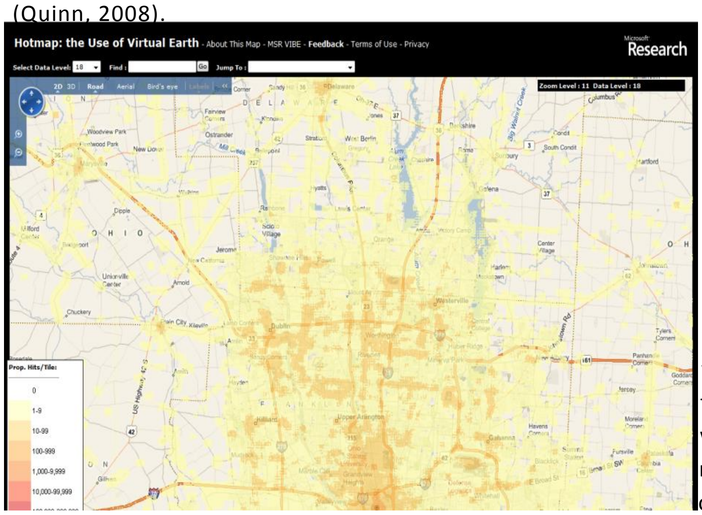
ArcGIS Server Cached Map Tile Tracking

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).


Hotmap: the Use of Virtual Earth - About This Map - MSR VIBE - Feedback - Terms of Use - Privacy
Microsoft Research
Select Data Level: 18
Find :
Go
Jump To :
2D 3D Road Aerial Bird's eye Labels
Zoom Level : 11 Data Level : 18
Prop. Hits/Tile:
0
1-9
10-99
100-999
1,000-9,999
10,000-99,999
O H I O

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 caching? It would be nice to know what tiles people are viewing and it would be even better if a model could display 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 are less desirable.

The purpose of this document is to describe how to implement a Descriptive Analysis Model so that organizations can track cached map tile usage using ArcGIS Server and Geoprocessing. Here is a link to an active tile-tracking site, 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.

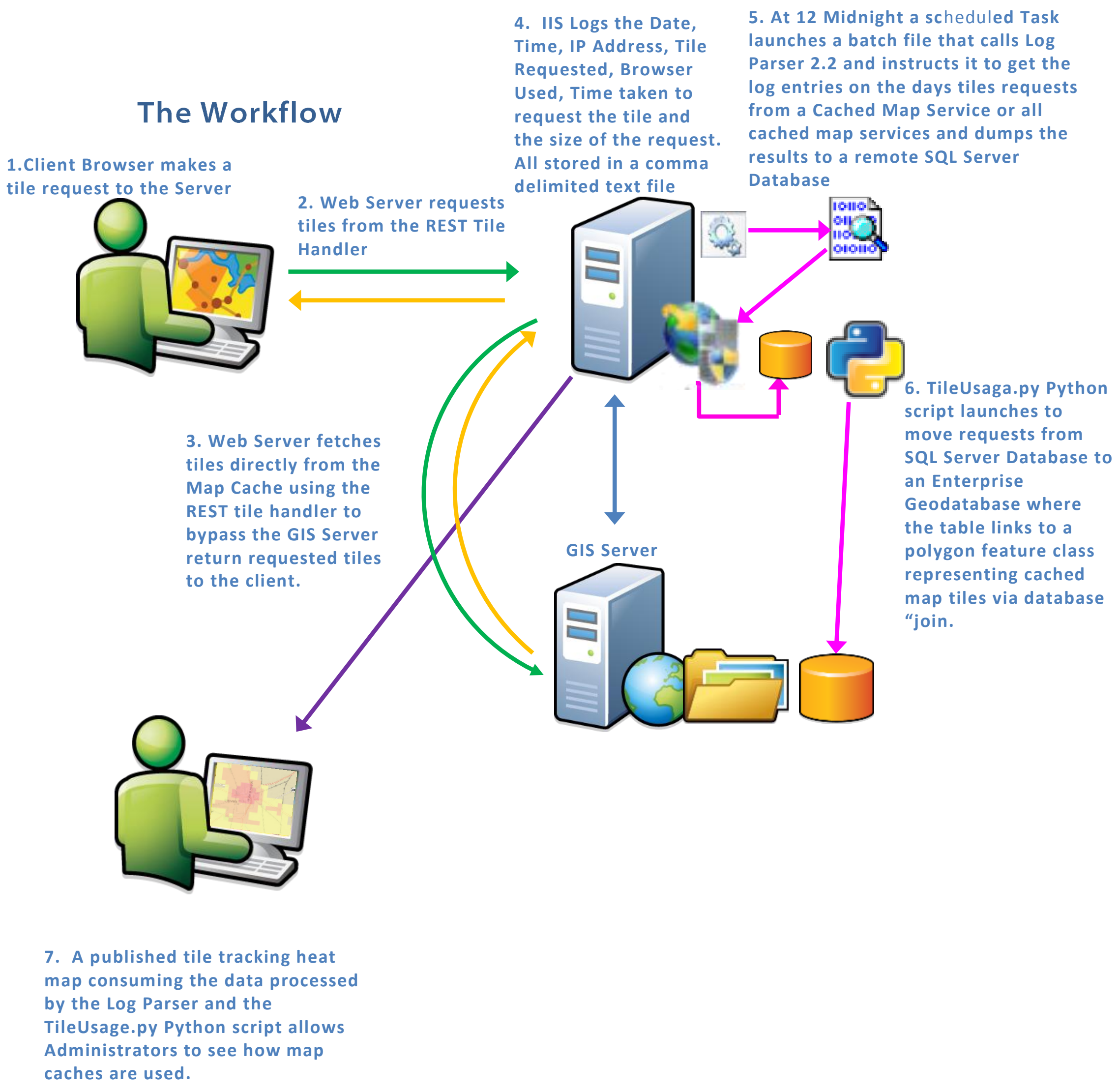
Requirements

The following is a list of software and tools used to create the tile tracking workflow used by ArcGIS Server. While this specific workflow utilizes Microsoft components for the web server, database logging, and scheduled task execution, components like Apache Tomcat web server, Log4J database logging, or JDK 1.4 database logging can work as well. Additionally, relational databases supported by ESRI used for data logging are acceptable. Successful tests of this workflow took place on a Windows 7 64-bit test environment and Windows Server 20008 32-bit production environment.

1. Administrative Privileges on the Web Server, ArcGIS Server
2. DBO privileges to the database.
3. ArcGIS Server 10 Standard or higher
    a. Windows Server 2008 Service Pack 2
    b. IIS 7.0
    c. Batch File and DOS
    d. Task Scheduler 1.0
4. ArcInfo 10
    a. Windows Server 2008 Service Pack 2
5. ArcPy Python Module
6. Microsoft SQL Express 2008
    a. SQL Server Native Client
7. ArcSDE Enterprise, Workgroup or Personal Edition
8. Microsoft Log Parser 2.2
9. AGSSOM v10

The key component to this workflow is Log Parser 2.2 Database Logging. It easily parses IIS logs, which is an unpleasant comma delimited text file. Microsoft develops Log Parser as an un-supported application and no problems occurred while working with it in Windows Server 2008. Log Parser allows administrators to execute SQL statements that extract any portion of the log files. 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 execute from command line or batch file, which in turn can execute from Windows Task Scheduler. If you are working in a 64-bit environment anytime the instructions say to locate c:\program files\, replace the line with c:\program files (x86)\ as this process runs in 32-bit mode.

The Workflow



1.Client Browser makes a tile request to the Server

2. Web Server requests tiles from the REST Tile Handler

3. Web Server fetches tiles directly from the Map Cache using the REST tile handler to bypass the GIS Server return requested tiles to the client.

4. IIS Logs the Date, Time, IP Address, Tile Requested, Browser Used, Time taken to request the tile and the size of the request. All stored in a comma delimited text file

5. At 12 Midnight a scheduled Task launches a batch file that calls Log Parser 2.2 and instructs it to get the log entries on the days tiles requests from a Cached Map Service or all cached map services and dumps the results to a remote SQL Server Database

6. TileUsaga.py Python script launches to move requests from SQL Server Database to an Enterprise Geodatabase where the table links to a polygon feature class representing cached map tiles via database "join.

7. A published tile tracking heat map consuming the data processed by the Log Parser and the TileUsage.py Python script allows Administrators to see how map caches are used.

Tile Tracking Setup Instructions

Step 1

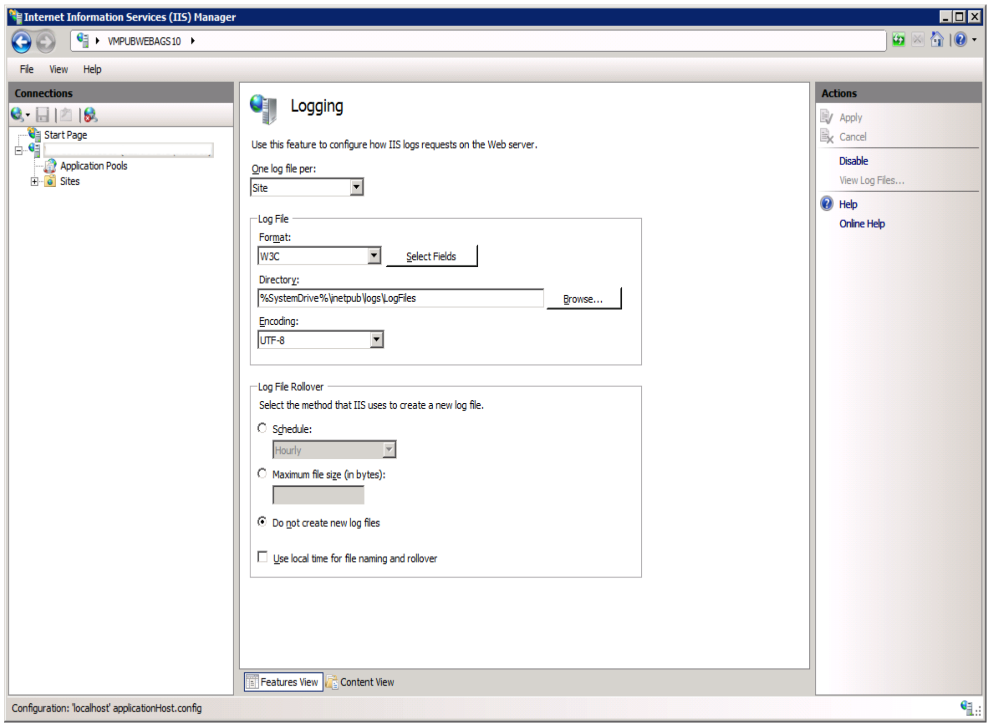
Download and install Log Parser 2.2 on the Web Server (IIS). Install Log Parser or the data logging software of your choice on the reverse proxy server if you want to identify the requesting IP address to determine what geographical region of the world the request is coming from.

Step 2 (Optional IIS Configuration Please Read)

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.



Logging

Use this feature to configure how IIS logs requests on the Web server.

One log file per: Site

Log File

Format: W3C [Select Fields]

Directory: %SystemDrive%\inetpub\logs\LogFiles [Browse...]

Encoding: UTF-8

Log File Rollover

Select the method that IIS uses to create a new log file.

- ○ Schedule: Hourly
- ○ Maximum file size (in bytes):
- ● Do not create new log files

☐ Use local time for file naming and rollover

Configuration: 'localhost' applicationHost.config

Step 3

Create the IISLogsLP Database on the server where SQL Server resides. For the prototype of this project, SQL Express, installed on the web server (IIS) in conjunction with Log Parser 2.2 managed the data logging. Before continuing, you need to make sure that you are an administrator of the database or can work with an account that has database owner privileges.

Browse to the folder .\TileTracking\SQL\ and open the CreateIISLogsLP_DB.sql with a text editor like notepad.

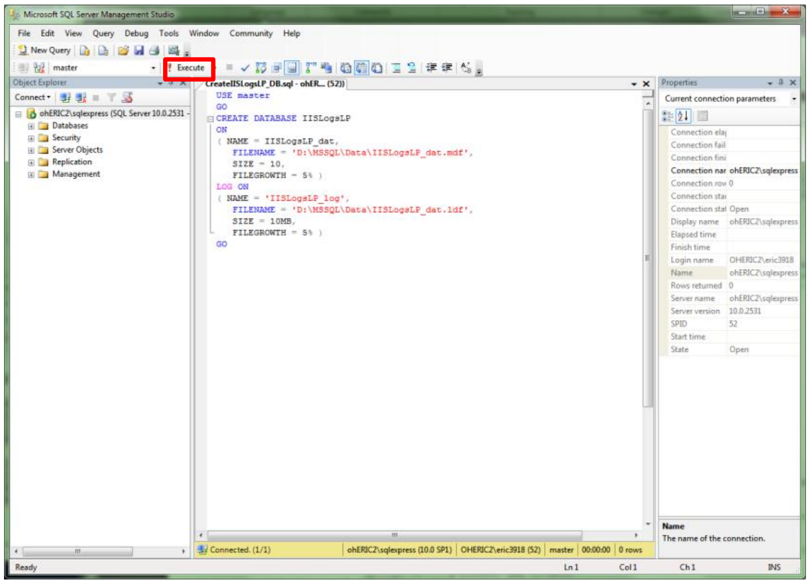
```
USE master
GO
CREATE DATABASE IISLogsLP
ON
( NAME = IISLogsLP_dat,
   FILENAME = 'D:\MSSQL\Data\IISLogsLP_dat.mdf',
   SIZE = 10,
   FILEGROWTH = 5% )
LOG ON
( NAME = 'IISLogsLP_log',
   FILENAME = 'D:\MSSQL\Data\IISLogsLP_dat.ldf',
   SIZE = 10MB,
   FILEGROWTH = 5% )
GO
```

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.


Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query
master
Execute
Object Explorer
Connect
ohERIC2\sqlexpress (SQL Server 10.0.2531 -
Databases
Security
Server Objects
Replication
Management
CreateIISLogsLP_DB.sql - ohER... (52))
Properties
Current connection parameters
Name
The name of the connection.
Ready
Ln 1
Col 1
Ch 1
INS

```
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
```

Connection ela	
Connection fail	
Connection fini	
Connection nar	ohERIC2\sqlexpress
Connection row	0
Connection sta	
Connection stat	Open
Display name	ohERIC2\sqlexpress
Elapsed time	
Finish time	
Login name	OHERIC2\eric3918
Name	ohERIC2\sqlexpress
Rows returned	0
Server name	ohERIC2\sqlexpress
Server version	10.0.2531
SPID	52
Start time	
State	Open

Connected. (1/1) | ohERIC2\sqlexpress (10.0 SP1) | OHERIC2\eric3918 (52) | master | 00:00:00 | 0 rows

Step 4

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. To locate the fields that can be imported, open the IIS log file and look for the line "#Fields:" This line contains all of the fields that IIS logs. 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. Finally, you will be selecting records that match the map cache service you entered based on comments found in the right margin. The final scheduled task will only request the records from the last 24 hours for the chosen map cache. The command line provided above will locate all tile hits that may currently exist.

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.

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
```

Special Note

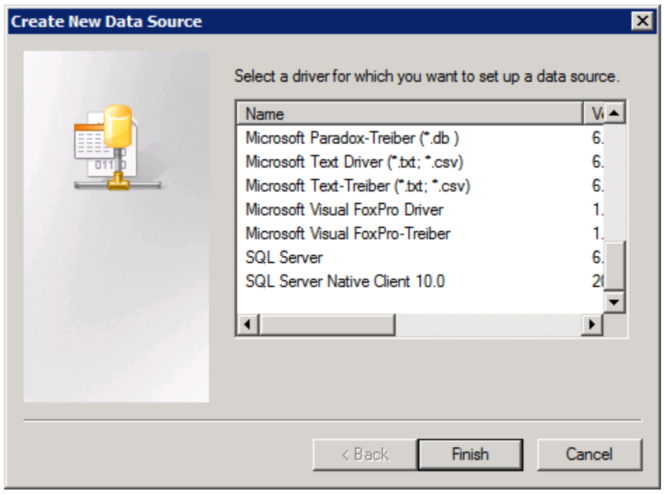
The version of SQL server you have determines the SQL Native Client Drivers you can call. The easiest way to determine this information: Start | Administrative Tools | Data Sources (ODBC Drivers) | Click the Continue button on the User Account Control Dialog *this requires Administrative Privileges* | 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 |

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.

Click Cancel to close the dialog



Select a driver for which you want to set up a data source.

Name	V
Microsoft Paradox-Treiber (*.db)	6.
Microsoft Text Driver (*.txt; *.csv)	6.
Microsoft Text-Treiber (*.txt; *.csv)	6.
Microsoft Visual FoxPro Driver	1.
Microsoft Visual FoxPro-Treiber	1.
SQL Server	6.
SQL Server Native Client 10.0	2(

Step 5

Open a command prompt.

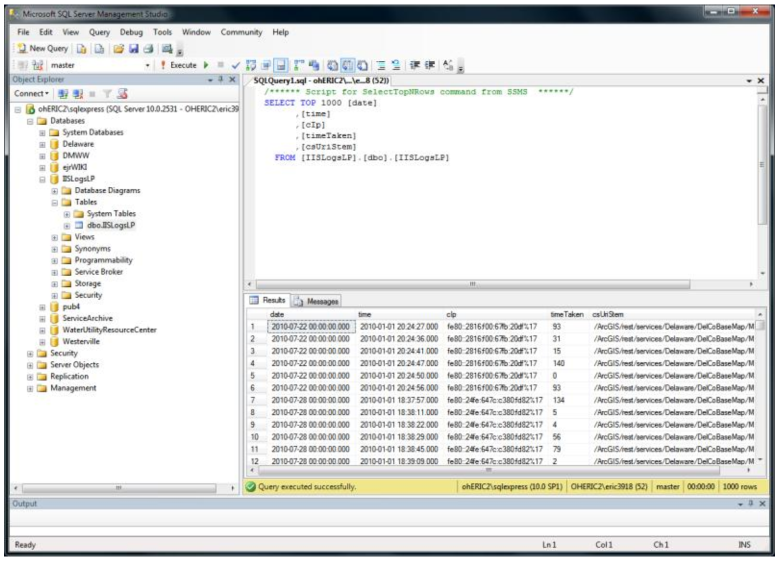
Start | Run| Type "CMD" in the command prompt | Enter

Copy the command we modified in Step 4, which is actually two lines of code, 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 the lines

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.



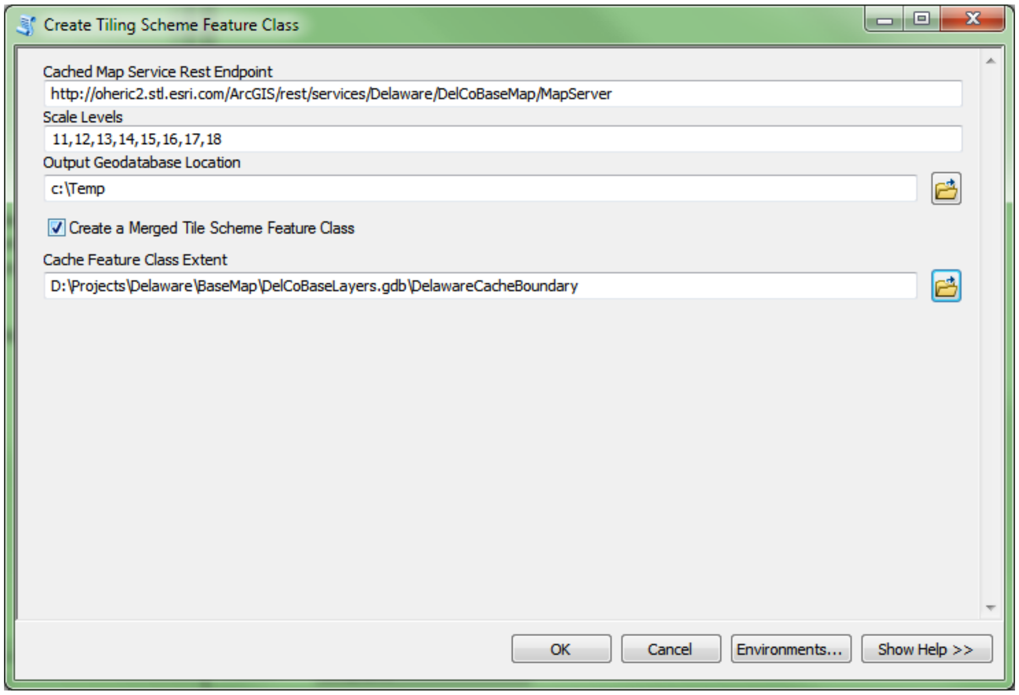
```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [date]
      ,[time]
      ,[cIp]
      ,[timeTaken]
      ,[csUriStem]
  FROM [IISLogsLP].[dbo].[IISLogsLP]
```

	date	time	cIp	timeTaken	csUriStem
1	2010-07-22 00:00:00.000	2010-01-01 20:24:27.000	fe80::2816:f00:67fb:20df%17	93	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
2	2010-07-22 00:00:00.000	2010-01-01 20:24:36.000	fe80::2816:f00:67fb:20df%17	31	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
3	2010-07-22 00:00:00.000	2010-01-01 20:24:41.000	fe80::2816:f00:67fb:20df%17	15	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
4	2010-07-22 00:00:00.000	2010-01-01 20:24:47.000	fe80::2816:f00:67fb:20df%17	140	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
5	2010-07-22 00:00:00.000	2010-01-01 20:24:50.000	fe80::2816:f00:67fb:20df%17	0	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
6	2010-07-22 00:00:00.000	2010-01-01 20:24:56.000	fe80::2816:f00:67fb:20df%17	93	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
7	2010-07-28 00:00:00.000	2010-01-01 18:37:57.000	fe80::24fe:647c:c380:fd82%17	134	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
8	2010-07-28 00:00:00.000	2010-01-01 18:38:11.000	fe80::24fe:647c:c380:fd82%17	5	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
9	2010-07-28 00:00:00.000	2010-01-01 18:38:22.000	fe80::24fe:647c:c380:fd82%17	4	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
10	2010-07-28 00:00:00.000	2010-01-01 18:38:29.000	fe80::24fe:647c:c380:fd82%17	56	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
11	2010-07-28 00:00:00.000	2010-01-01 18:38:45.000	fe80::24fe:647c:c380:fd82%17	79	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M
12	2010-07-28 00:00:00.000	2010-01-01 18:39:09.000	fe80::24fe:647c:c380:fd82%17	2	/ArcGIS/rest/services/Delaware/DelCoBaseMap/M

Query executed successfully. | ohERIC2\sqlexpress (10.0 SP1) | OHERIC2\eric3918 (52) | master | 00:00:00 | 1000 rows

Step 6

Download and install Create Tiling Scheme Feature Class python script and run it to create the Tiling Scheme geodatabase. This script will create a feature class representation 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.



Cached Map Service Rest Endpoint
http://oheric2.stl.esri.com/ArcGIS/rest/services/Delaware/DelCoBaseMap/MapServer
Scale Levels
11,12,13,14,15,16,17,18
Output Geodatabase Location
c:\Temp
[✓] Create a Merged Tile Scheme Feature Class
Cache Feature Class Extent
D:\Projects\Delaware\BaseMap\DelCoBaseLayers.gdb\DelawareCacheBoundary

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

Step 7

Create an SDE Geodatabase and load the Merged Tile Scheme Feature Class to facilitate tile tracking. While prototyping this workflow, ArcSDE Workgroup edition tracked the individual tiles and process the results of the information captured by Log Parser 2.2. You can use ArcSDE Enterprise but the following instructions outline how to create the database using ArcSDE Workgroup Edition. The database resides on the Web Server with ArcGIS Server.

1. Open ArcCatalog and locate the Database Servers Node in the Catalog Tree

2. Open the Database Servers node and click on your database instance connection. The name will be Server_Instance.gds. If this does not exist, double click Add Database Server. Enter the "Server Host Name\sqlexpress" and click ok.
3. Right click on the Server_Instance.gds connection and choose Permissions > Add User and Add the ArcGIS SOC Account as well as your username. If the machine you are working on does not have ArcGIS Server installed you will have to manually add the ArcGISSOC account as well as a password that matches the ArcGISSOC account on your ArcGIS Server.
4. Right click on your Server_Instance.gds connection file and choose "Connect."
5. Right click on the Server_Instance.gds and choose "New Geodatabase"
    a. Geodatabase Name should be TileTracking
    b. Location –Can be the default however if your machine has multiple hard drives or a partitioned hard drive a good practice is to store the database on a drive other than C: which is typically reserved for system files.
    c. Initial Size 10 MB
    d. Ok
6. Once the TileTracking Geodatabase is created right click on it and choose Administration > Permissions > ensure that the ArcGISSOC account is visible and has a minimum of read only permission.
7. Locate the TileScheme File geodatabase created in step 6. Expand the node to list all of the feature classes. Locate the CacheMapTilingScheme feature class. You can rename the feature class to match the name of the service that you are tracking. The feature class created for this workflow, DelCoMapTilingScheme, is tracking a service called DelCo.
8. Right click on renamed CacheMaptilingScheme feature class and choose copy.
9. Right click on the Tile Tracking Enterprise database in the Database Servers Node, choose Paste, and click OK to confirm the data transfer.
10. Right click on the Tile Tracking Enterprise database and choose "Save Connection"
    a. Open the Database Connections Node below Database Servers.
    b. Look for Connection to Server_Instance.sde. Server being the name of your server's host name. Right click and choose "Rename" Name the connection as follows: DC@HostName@TileTracking.sde. This means DirectConnect@HostName or ServerName @Database Name.
    c. Copy the Renamed SDE connection and paste it into the DataConnections Folder created when you unzipped the TileTracking Workflow.
    d. Make sure that the ArcGISSOC account has access to this folder. The ArcGISSOC account needs to have a minimum of "Read Only" privileges

Step 8

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 >

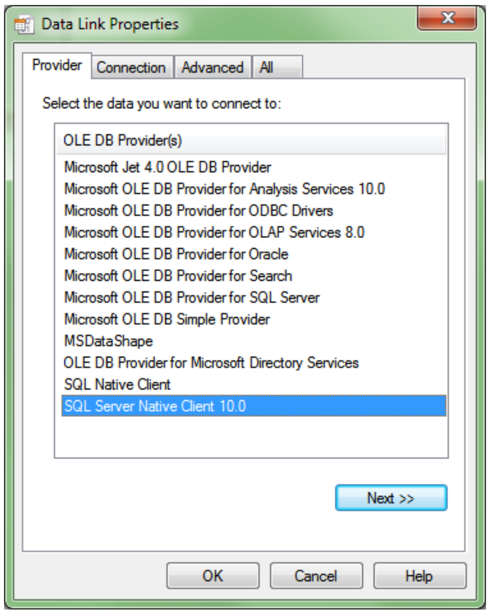
1. Enter a Server Name: hostname\Database Instance

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

3. Select a Database: IISLogsLP

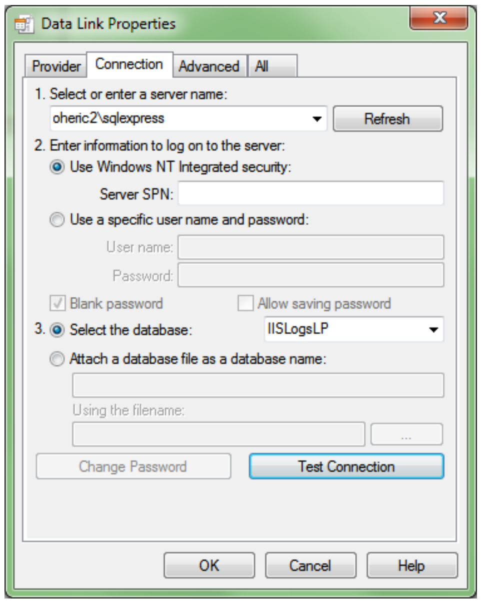
Click the Test Connection Button.

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



Select the data you want to connect to:

OLE DB Provider(s)
Microsoft Jet 4.0 OLE DB Provider
Microsoft OLE DB Provider for Analysis Services 10.0
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services 8.0
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services
SQL Native Client
SQL Server Native Client 10.0



1. Select or enter a server name: oheric2\sqlexpress

2. Enter information to log on to the server:
(●) Use Windows NT Integrated security:
Server SPN:
( ) Use a specific user name and password:
User name:
Password:
[✓] Blank password [ ] Allow saving password

3. (●) Select the database: IISLogsLP
( ) Attach a database file as a database name:
Using the filename:

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 9

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 8.

Update the next line, which is the location of the SDE connection file saved in step 7.

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 7. 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 should use for the user.

Step 10

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

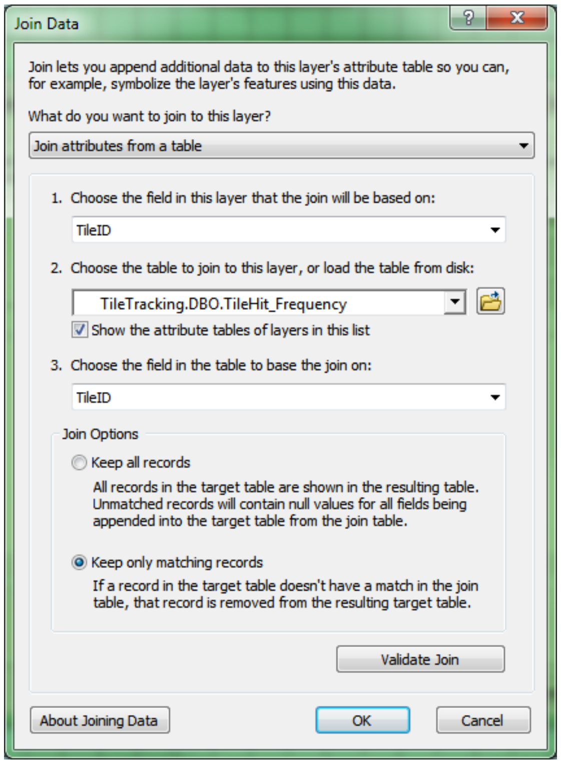
```
C:\Python26\ArcGIS10.0\python.exe "%Your
Path%\TileTracking\Python\TileUsage\TileUsage.py"
```

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 11

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:


Join Data

Join lets you append additional data to this layer's attribute table so you can, for example, symbolize the layer's features using this data.

What do you want to join to this layer?

Join attributes from a table

1. Choose the field in this layer that the join will be based on:

TileID

2. Choose the table to join to this layer, or load the table from disk:

TileTracking.DBO.TileHit_Frequency

[✓] Show the attribute tables of layers in this list

3. Choose the field in the table to base the join on:

TileID

Join Options

○ Keep all records

All records in the target table are shown in the resulting table. Unmatched records will contain null values for all fields being appended into the target table from the join table.

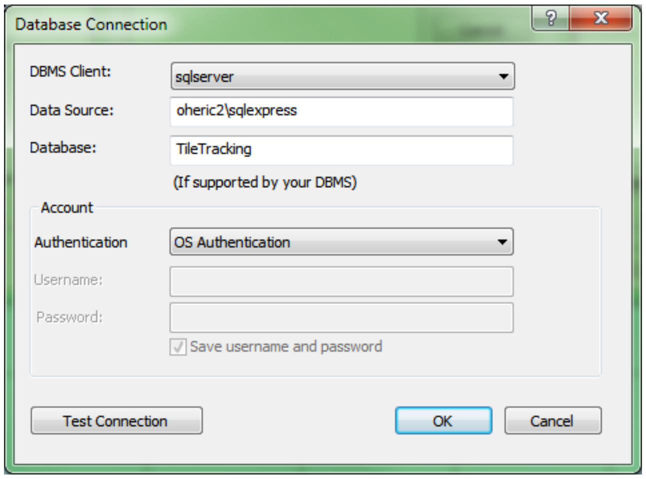
[✓ Keep only matching records]

If a record in the target table doesn't have a match in the join table, that record is removed from the resulting target table.

3. Click OK and Ok on the Properties Dialog

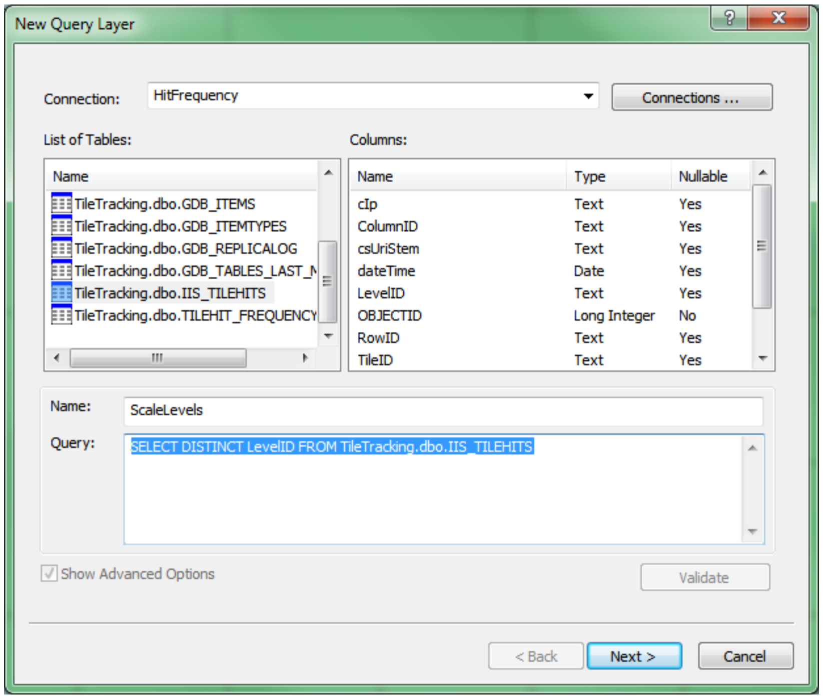
Notice the Join Options are set to use Keep only Matching Records. This is critical to the overall performance of the web application used to monitor which tile usage. The feature class used to model the cache for Delaware County, Ohio contains over 500,000 tiles and by "Keeping only matching records" 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: I used Workgroup SDE so the data source was Host\Instance which in my case was oheric2\sqlexpress
    f. Type in the name of the database which is TileTracking
    g. In the case of Workgroup SDE, 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


Database Connection

DBMS Client: sqlserver
Data Source: oheric2\sqlexpress
Database: TileTracking
(If supported by your DBMS)
Account
Authentication: OS Authentication
Username:
Password:
Save username and password

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.IIS_TILEHITS table
m. In the query box type the following SQL Query: SELECT DISTINCT LevelID FROM TileTracking.dbo.IIS_TILEHITS
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



New Query Layer

Connection: HitFrequency

List of Tables:

Name
TileTracking.dbo.GDB_ITEMS
TileTracking.dbo.GDB_ITEMTYPES
TileTracking.dbo.GDB_REPLICALOG
TileTracking.dbo.GDB_TABLES_LAST_M
TileTracking.dbo.IIS_TILEHITS
TileTracking.dbo.TILEHIT_FREQUENCY

Columns:

Name	Type	Nullable
cIp	Text	Yes
ColumnID	Text	Yes
csUriStem	Text	Yes
dateTime	Date	Yes
LevelID	Text	Yes
OBJECTID	Long Integer	No
RowID	Text	Yes
TileID	Text	Yes

Name: ScaleLevels

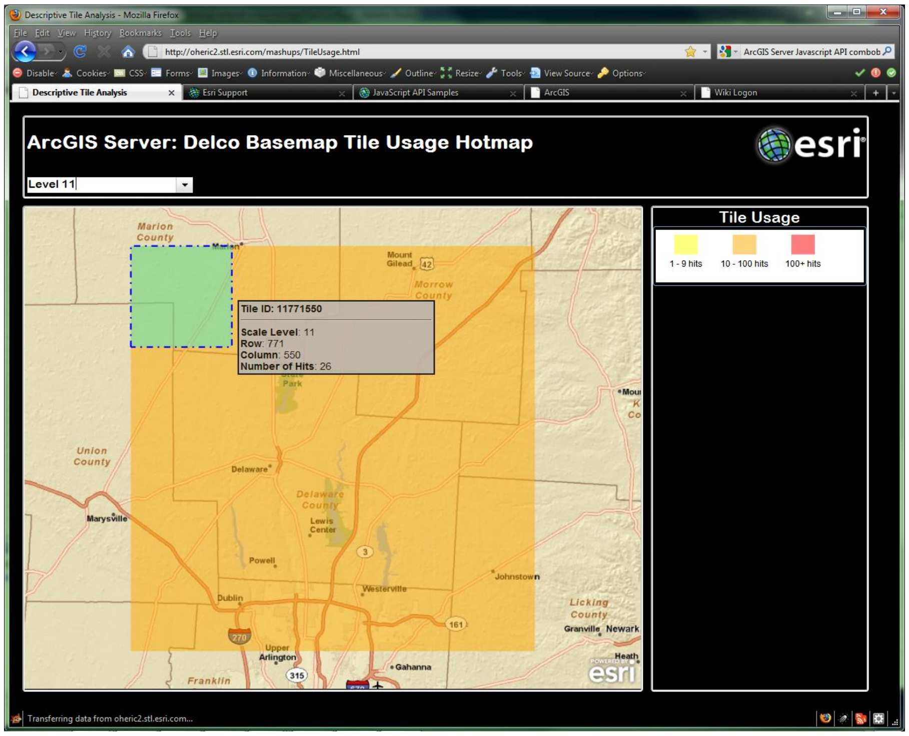
Query: SELECT DISTINCT LevelID FROM TileTracking.dbo.IIS_TILEHITS

5. Save the Map as CacheUsageHeatMap
6. Open the Map Service Publishing Toolbar.
7. Choose the Save Map Service Definition Button
8. Save the MSD in the same directory as the MXD
9. Choose the Publish to ArcGIS Server Button to publish the MSD as a Map Service.

Step 12

This workflow provides a custom application mashup that combines the World Streetmap from ArcGIS Online with the service published in step 11 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 11.

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 11. 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


Descriptive Tile Analysis - Mozilla Firefox
http://oheric2.stl.esri.com/mashups/TileUsage.html
ArcGIS Server: Delco Basemap Tile Usage Hotmap
Level 11
Tile Usage
1 - 9 hits
10 - 100 hits
100+ hits
Tile ID: 11771550
Scale Level: 11
Row: 771
Column: 550
Number of Hits: 26
Transferring data from oheric2.stl.esri.com...

Step 13

In order to update the results displayed by the application in step 12, you need to stop the CacheUsageHeatMap web service in order to update the frequency tables with the latest set of records from the last 24 hours. To do this, use the AGSSOM v10 utility found on the ArcGIS Resource center. This utility allows administrators to start, stop, and pause individual services or all services via command line.

Download and install the AGSSOMv10 utility and note the path for Step 14.

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\TileTracking.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 13 lines of code, which also includes comments.

```
Rem Stop Tileusage Service so that the IIS Logs can be refreshed with the
previous days activities
cd /D "C:\ags_share\TileTracking\AGSSOMv10.0"
AGSSOM.exe -x delaware/CacheUsageHeatmap

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
echo on
C:\Python26\ArcGIS10.0\python.exe
"C:\ags_share\TileTracking\Python\TileUsage\src\TileUsage.py"
echo off

Rem Restart Tile Usage Map Service
cd /D "C:\ags_share\TileTracking\AGSSOMv10.0"
AGSSOM.exe -s delaware/CacheUsageHeatmap
```

Comment [EJR10]: Change the directory of the AGSSOMv10.0 tool to point to the location you installed it in.

Comment [EJR11]: Stopping the CacheUsageHeatMap Service

Comment [EJR12]: Changing the directory to the Log Parser Home Directory

Comment [EJR13]: 2 %% signs required for DOS bat to interpret the % as a partial string wildcard. This was 2 days of pain and suffering

Comment [EJR14]: 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 [EJR15]: Get the last 24 hours of records.

Comment [EJR16]: 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 [EJR17]: Changing directories to the python directory

Comment [EJR18]: Calling the Tile Usage Script

Comment [EJR19]: Make sure the path name is correct and is pointing to the directory

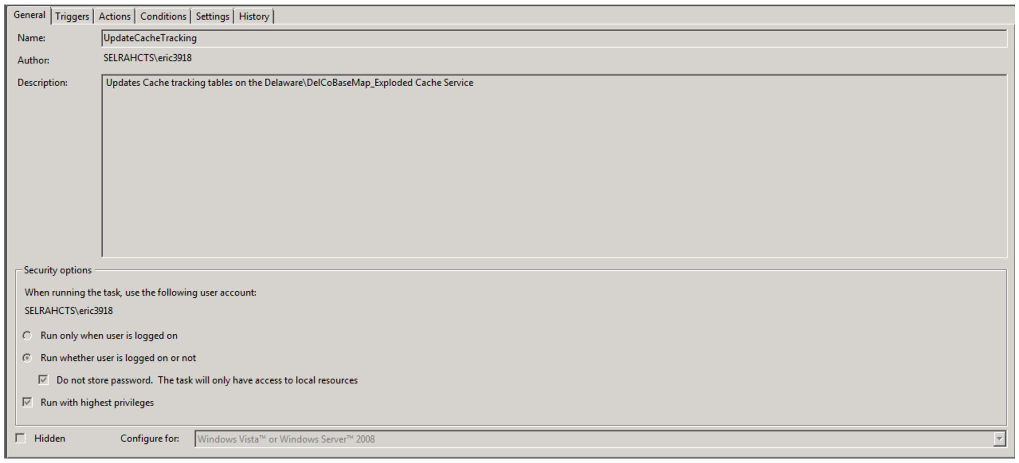
Comment [EJR20]: Start the updated CacheUsageHeatmap back up. Notice that the service is inside a folder called Delaware. If your service is not in a folder then you won't have anything in front of your service name.

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.

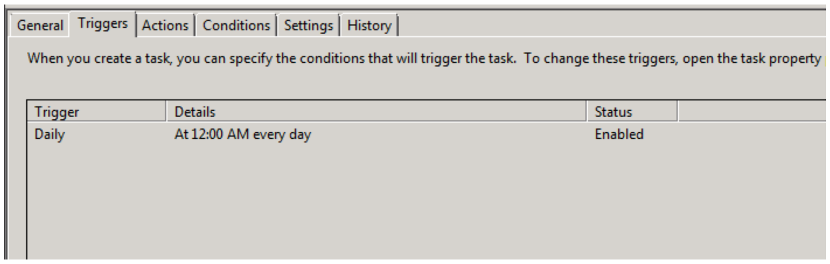


Name: UpdateCacheTracking
Author: SELRAHCTS\eric3918
Description: Updates Cache tracking tables on the Delaware\DelCoBaseMap_Exploded Cache Service

Security options
When running the task, use the following user account:
SELRAHCTS\eric3918
( ) Run only when user is logged on
(●) Run whether user is logged on or not
[✓] Do not store password. The task will only have access to local resources
[✓] Run with highest privileges

[ ] Hidden Configure for: Windows Vista™ or Windows Server™ 2008

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



When you create a task, you can specify the conditions that will trigger the task. To change these triggers, open the task property

Trigger	Details	Status
Daily	At 12:00 AM every day	Enabled

Click the Actions Tab browse to the TileTracking.bat, file configured in step 9.

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