GenerateGeotiffv1.m
Makegridv1.m

User Manuals

Using Matlab by MathWorks™

MATLAB Release: Matlab 7.8 (R2009a)

Nick Mortimer
Carlin Bowyer

Contents Page:

ACKNOWLEDGEMENTS: 1

CODE DESCRIPTIONS: 2

SECTION 1: USING GENERATE_GEOTIFFV1 3

STEP 1: SETTING UP YOUR COMPUTER TO USE GENERATEGEOTIFF.M: 3
STEP 2: SETTING UP GDAL FWTOOLS: 4
Downloading GDAL: 4
Installing GDAL: 7
STEP 3: SETTING THE FILE PATH TO THE BIN FILE: 10
STEP 4: RUNNING GENERATEGEOTIFF.M: 12
Data Acquisition: 12
Data Input to GenerateGeotiff.m: 14
a) Set the server: 14
b) Set the image coordinates: 14
c) Set the zoom: 16
d) Setting the image Date: 16
e) Enter Pathname: 16
f) Enter Filename: 17

SECTION 2: USING MAKEGRIDV1 18

STEP 1: SETTING UP YOUR COMPUTER TO USE MAKEGRID.M: 18
STEP 2: SETTING THE PATH TO THE BIN FILE: 18
STEP 3: RUNNING MAKEGRID.M: 18
Data Acquisition: 18
Allowable Spatial extents in MATLAB: 19
Data input into makegrid.m: 19
a) Set the server: 19
b) Set the image coordinates: 20
c) Set the zoom: 20
d) Setting the image Date: 21
e) Enter Pathname: 21
f) Enter Filename: 22
g) xblock and yblock: 22

APPENDICES: 24

APPENDIX1: 24
APPENDIX2: 25

Acknowledgements:

I would just like to thank Nick Mortimer from the CSIRO for programming both getTile.m and makeGeotiff.m. I would also like to thank Nick for his assistance in debugging the scripts and running the programs.

I would also like to thank Ryan Lowe, UWA for assisting in some of the MATLAB Script Debugging and general guidance throughout the project.

Thanks, also goes to Bryan Boruff UWA for his general guidance throughout the project, in particular with the image integration into ArcGIS, and the use of his laptop during the testing stage of makeGeotiff.m. Thanks also to the Australian National Network in Marine Science (ANNIMS) for providing the funding required allowing the project to be undertaken.

Code Descriptions:

GenerateGeotiff.m

GenerateGeotiff.m is a MATLAB script which can be used to attain image tiles from image sources such as Google Earth and Nearmap. The program then writes the images to Geotiff format, allowing the images to be recognised, used and modified in geospatial programs such as ESRI's ArcMAP. GenerateGeotiff.m in its current version (1.0) is primarily focused to attain images from Nearmap, however by changing the server which the program points to there is the potential to attain images from Google Earth.

GenerateGeotiff.m is a combination of the getTile.m and makeGeotiff.m MATLAB scripts. GenerateGeotiff.m uses getTile.m to attain the image from the image server. GenerateGeotiff.m then uses makeGeotiff.m to write that image file into a Geotiff format which can be subsequently used in geospatial programs.

getTile.m contains a date stamp or date string field, which allows the user to attain any of the dated images available on the Nearmap server. This allows temporal comparison between images of different dates. This is one of the reasons why this small program is quite powerful and useful.

makeGrid.m

The second program illustrated in this manual is makeGrid.m. makeGrid.m is essentially the same as GenerateGeotiff.m. makeGrid.m differs in that the program is used to break up large images into allotments of tiles which the user can specify the desired number in the x and y direction. (E.g. tiles in the x direction and 5 tiles in the y direction, totalling 25 tiles in the tile allotment. total). These allotments are beneficial as;

a) It increases the extent of image which can be attained using MATLAB. GenerateGeotiff.m can only attain images with a single array of ~600MB. makeGrid.m splits the array into portions (or smaller arrays) therefore allowing MATLAB to extend its total array capacity to around ~1.5GB. This is especially crucial when dealing with high resolution imagery.
b) When the image of a feature is >1.5MB, sections of the image can be attained by adjusting the top and bottom latitudes only, reducing the inconvenience of having to break up the image.
c) The user can define the geographic size of each allotment of tiles making up the larger image. This allows the user to have control over the computational sizes of images which the user attains.

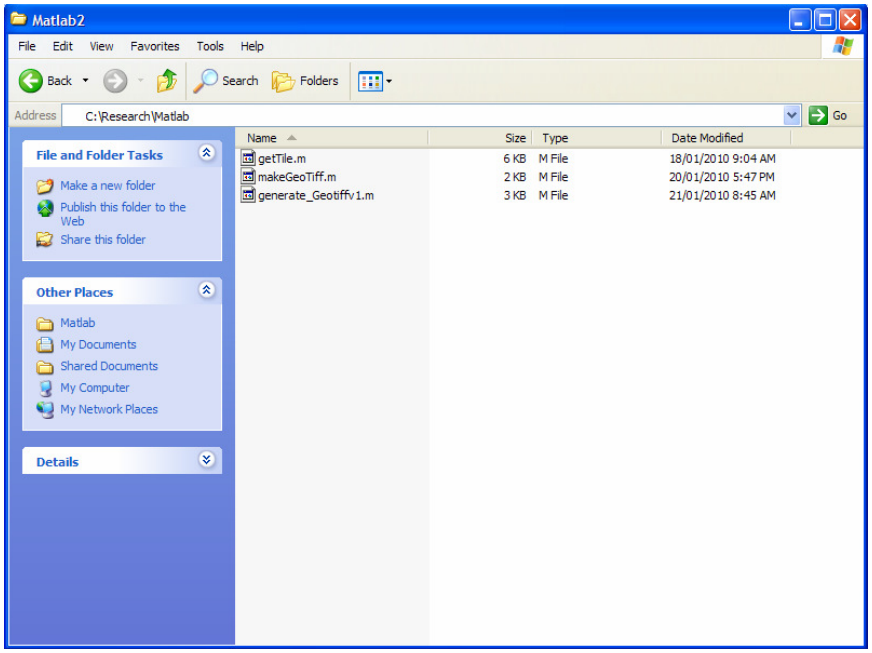
The user can determine which allotments of tiles are useful for feature analysis, allowing the user to import these into the GIS system and disregard any allotments of tiles which are not useful in the analysis. This also ensures that computing memory is not wasted on parts of the image which are not required. For example some tile allotments may be totally ocean, which is not required if the total image is being analysed for terrestrial purposes.

Section 1: Using generate_GeoTiffv1

Step 1: Setting up your computer to use GenerateGeotiff.m:

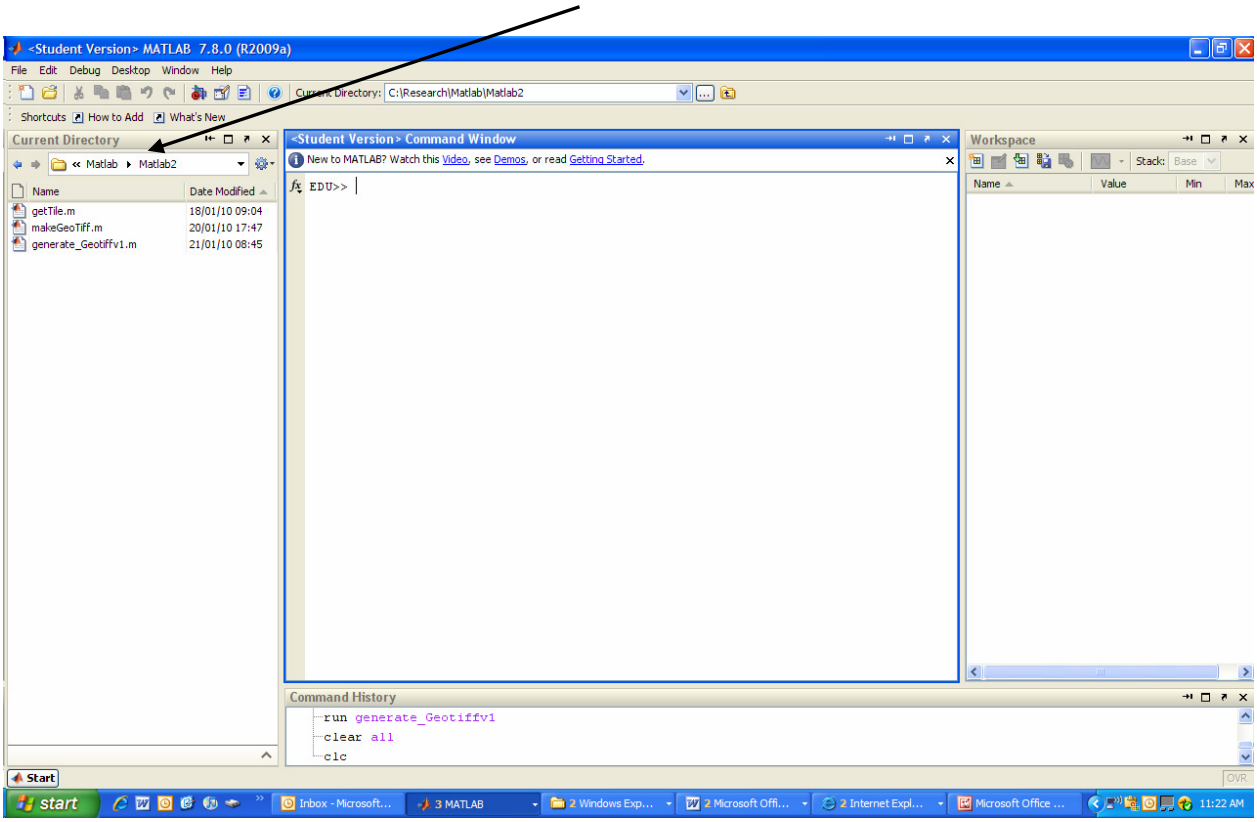
To enable use of GenerateGeotiff.m the computer must have both MATLAB and GDAL tools installed. MATLAB is a program by MathWorks™ which will need to be purchased, and installed like other software. This manual assumes that MATLAB is installed on a Windows computer.

Once MATLAB is installed, getTile.m, makeGeotiff.m, and GenerateGeotiff.m can be loaded into the MATLAB current directory. It is likely that you have all three files in a zipfile (generageGeotiffv1), so just extract these files from the zipfile and copy getTile.m , makeGeotiff.m, and GenerateGeotiff.m to the MALTAB "Current Directory" folder. You could also set the folder that you extracted these files to as the current directory in MATLAB. **Ensure that you keep a backup of each of the m files (I.e. Keep a backup of the Zip File) which is not in the MATLAB "current directory" in case the m files in the MATLAB "current directory" become corrupt.** Below is an example of the folder which the MATLAB "current directory" accesses to execute the scripts in order to run GenerateGeotiff.m .



Note: The file path to the current directory must not contain and files with spaces in them, such as "Documents and Settings". The best way to do this is to assign the "Current Directory" to a folder with a file path simular to that illustrated above (This is because GDAL Tools uses DOS command programming to make the Geotiff).

Once copying GenerateGeotiff.m, getTile.m, and makeGeotiff.m into the MATLAB "current directory" or setting the extracted zip file as the "current directory" in MATLAB, the "current directory" pane should look like simular to what is illustrated below.



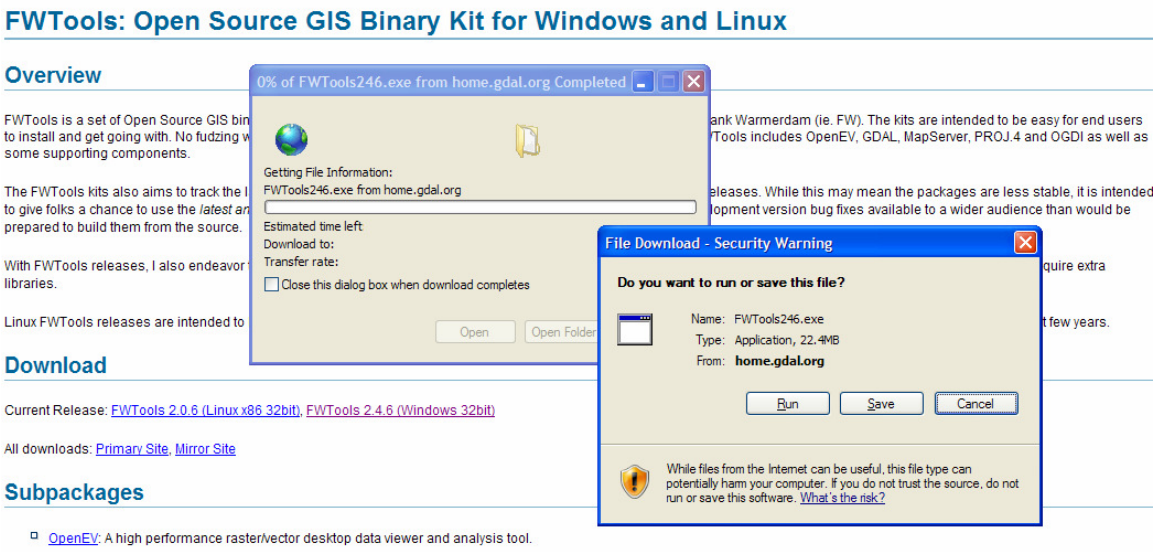
Step 2: Setting up GDAL FWTools:

GDAL is slightly more complex. GDAL is available for free download on the web, however requires to be installed through the Command Prompt (dos) on your computer. If GDAL (FWTools246 or simular) is installed on your computer proceed to step 3.

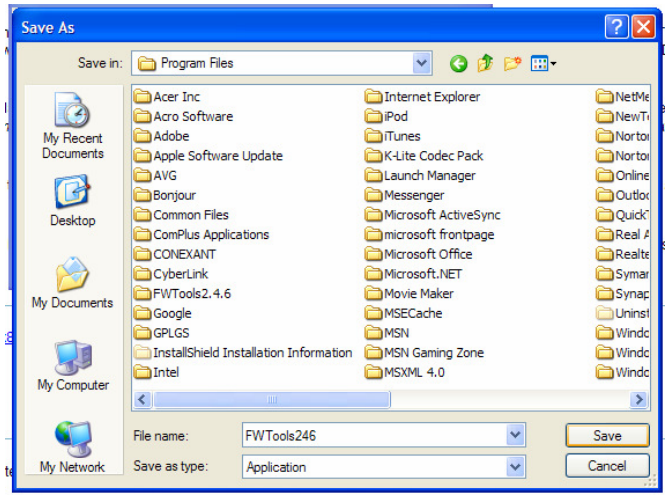
Downloading GDAL:

GDAL is available for download from:
http://fwtools.maptools.org/

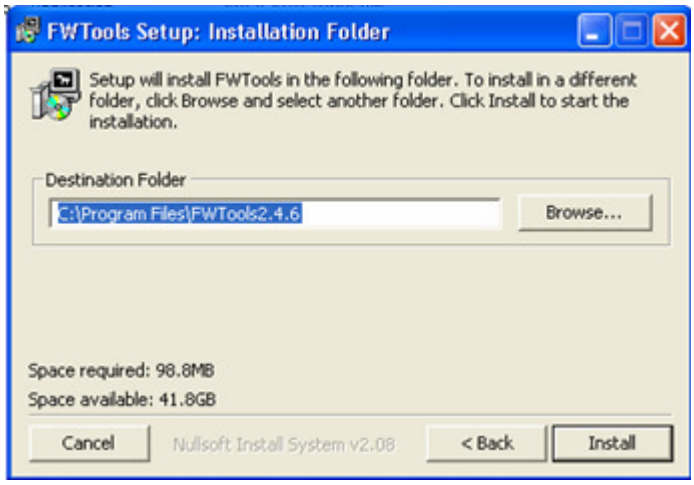
This page is titled "FWTools: Open Source GIS Binary Kit for Windows and Linux". Scroll down till the download section is reached. Download the current release for either Windows or Linux, depending on your operating system. This manual was constructed using a windows XP operating system. At the time this manual was written the current release filename for Windows was "FWTools 2.4.6 (Windows 32bit)". See Below:



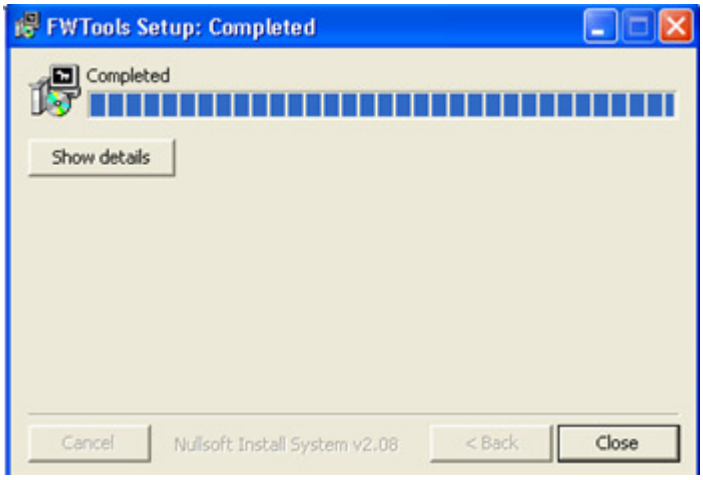
When it prompts, save this to your computer in the program files. See Below.



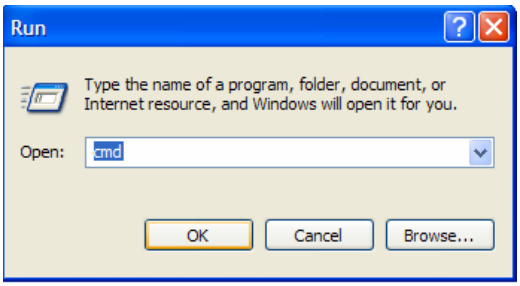
After the FWTools 2.4.6 (Windows 32bit) file is saved to your computer you will need to install it. To install double click on FWTools 2.4.6.exe, and go through the steps to install FWTools 2.4.6. The FWTools 2.4.6 Setup looks simular to the image below.



The default installation folder is FWTools 2.4.6 in the program files folder on the C drive. This can be changed however it is suggested that you just install FWTools 2.4.6 in its default location. Once FWTools 2.4.6 setup is completed the FWTools 2.4.6 setup will look simular to below.

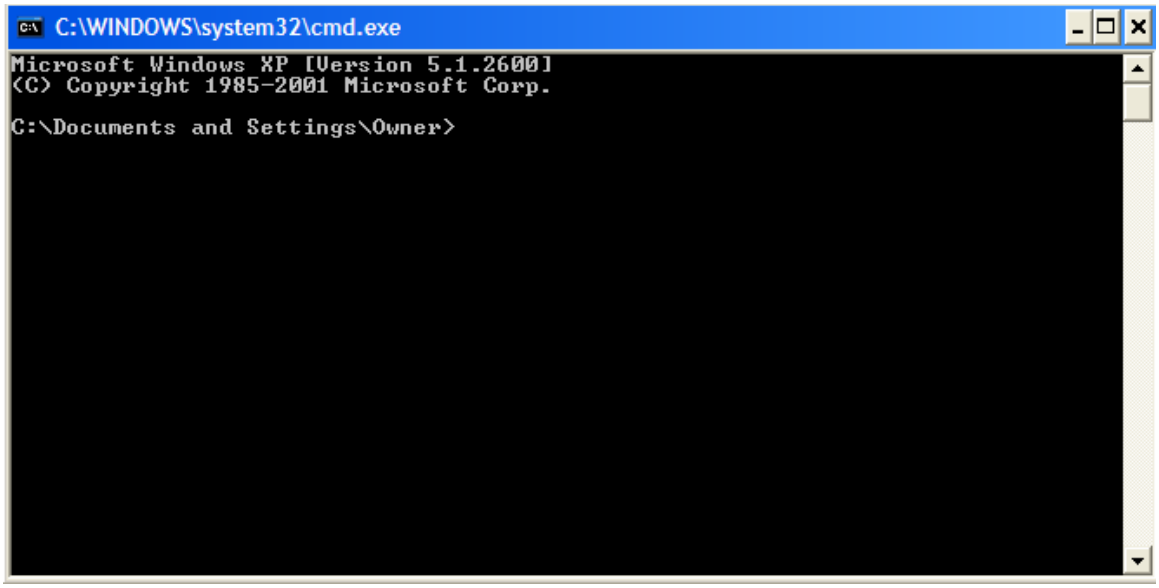


Once the FWTools246 is installed on your computer it can be set using the command prompts. To open the command window click start > Run. Enter “cmd” into the window, like shown below.

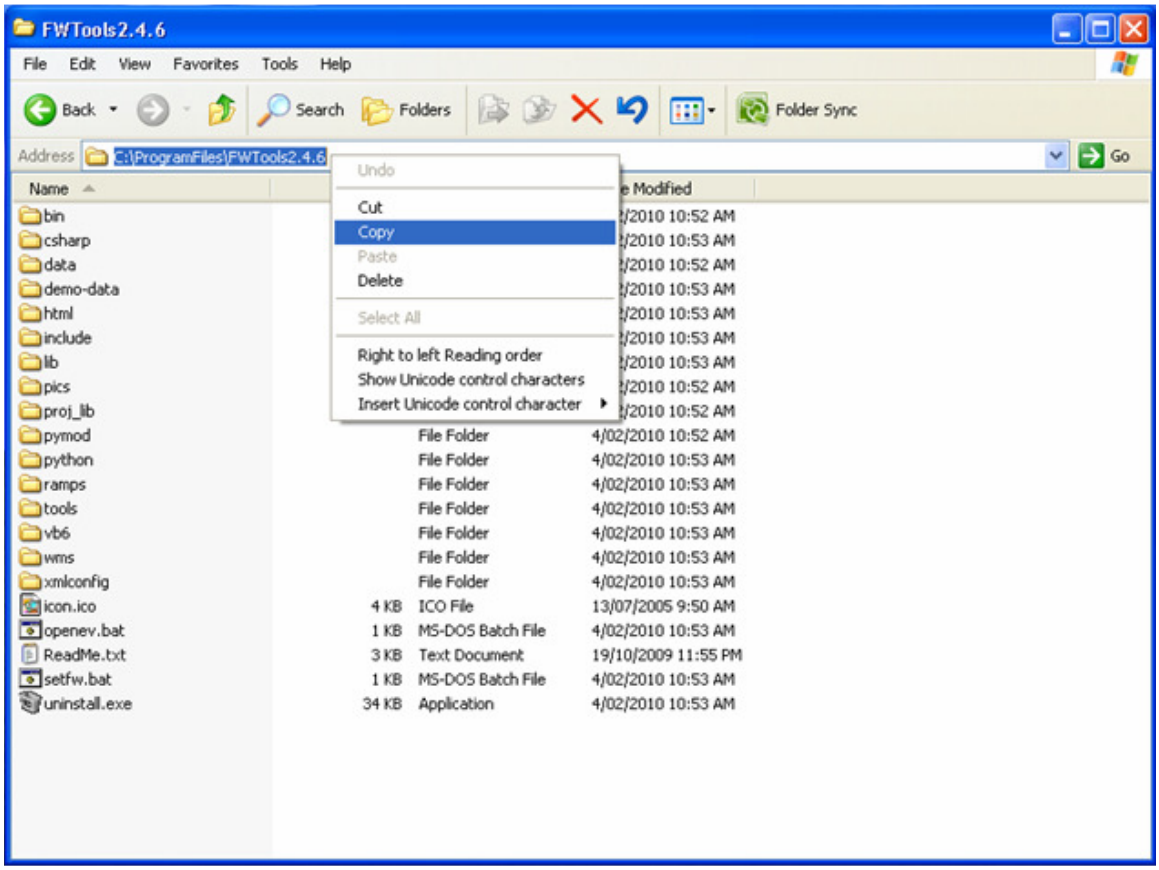


Installing GDAL:

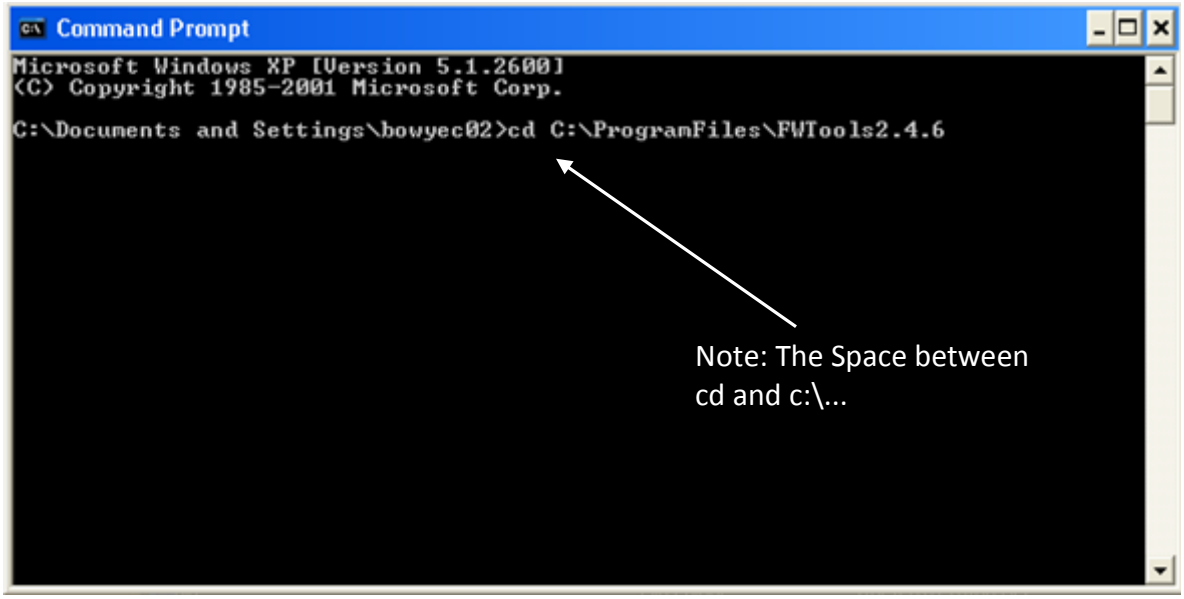
When opened the command prompt will look something like below.



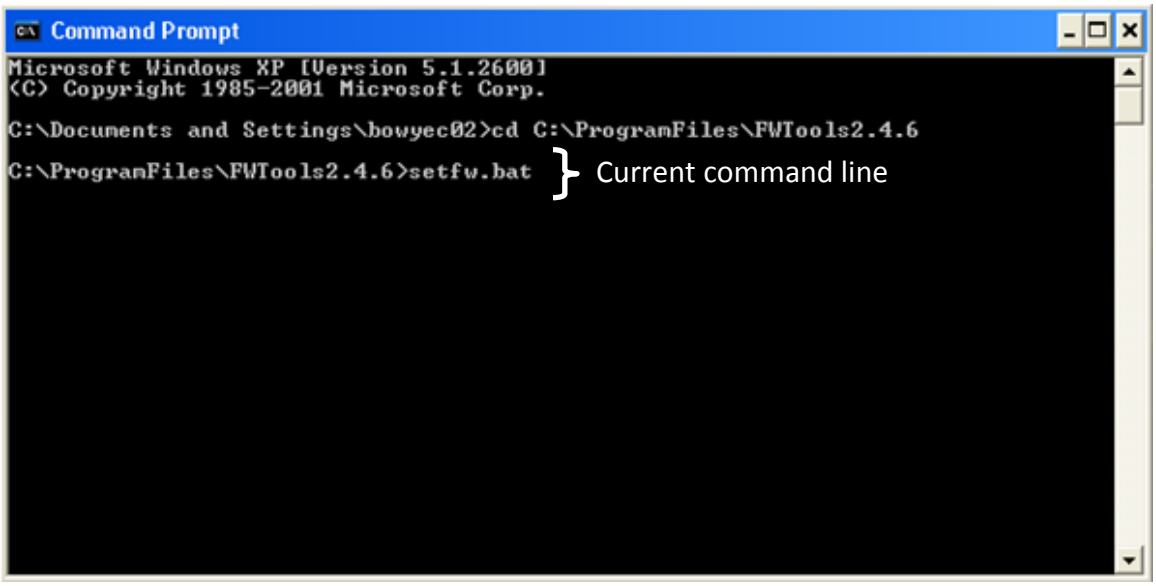
Once the command window is open, open My Computer (or simular) and navigate to where the FWTools246 file was installed in the previous section. Open FWTools 2.4.6 folder and copy the address as shown below. The file path is copied firstly for ease and secondly so that the command prompt can find install of FW Tools.



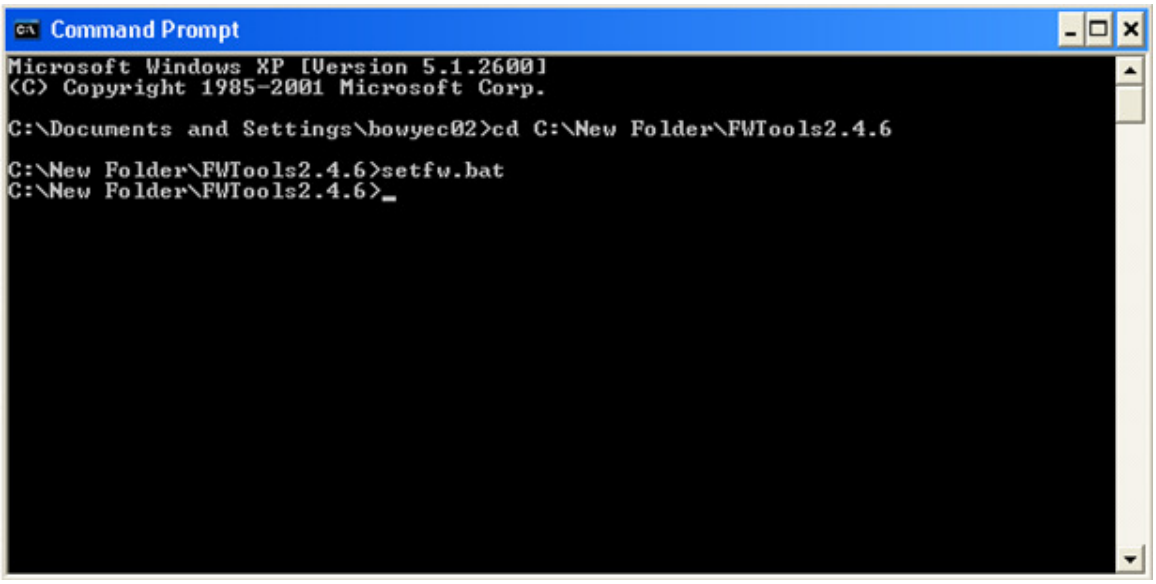
In the command prompt type “cd” and then place a space after “cd”. After the space copy the address for the bin file in via “right click” and selecting paste. Then press enter. The command window should look simular to below.



On the current command line type “set” and then press the tab key on the keyboard. This completes the command line to read setfw.bat as shown below. After this press the enter key on the keyboard and GDAL and FWTools should be installed on your machine.

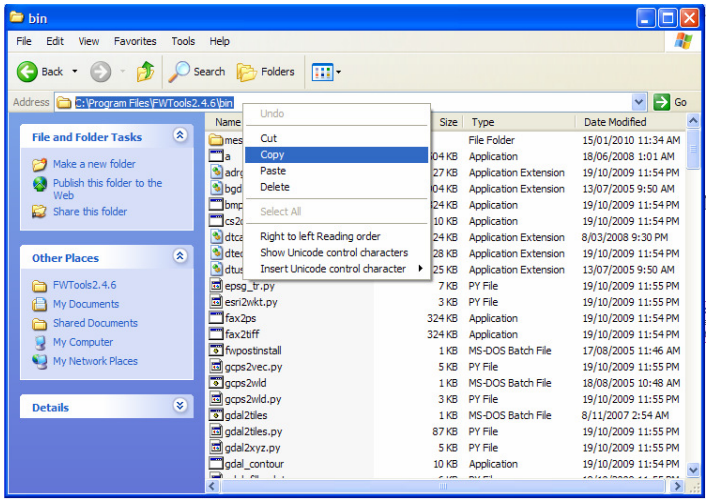


Once FWTools2.4.6 is installed on your machine there will be no message saying successful installation or simular; however if successful installation there will be no error message in the command prompt. The command prompt will look simular to below after successful installation.



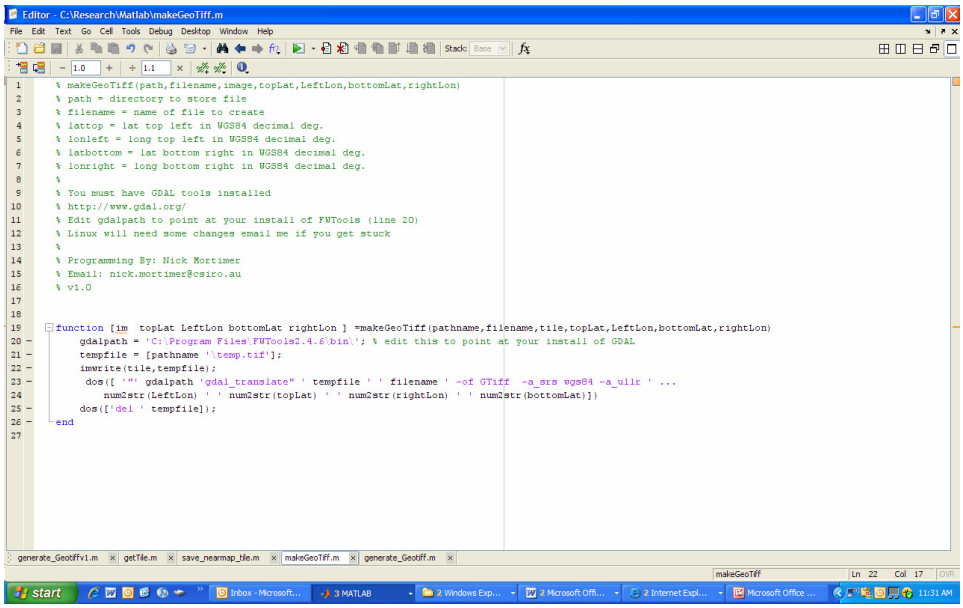
Step 3: Setting the file path to the Bin File:

Once FWTools is installed, open the FWTools folder and navigate to the bin file (As undertaken in the previous section) and copy the file path to the bin folder. The bin file is a file required by makeGeotiff.m for the conversion of the acquired image to Geotiff format. The bin file path will need to be added to the makeGeotiff program the first time it is used. See below for an example of how to copy the bin folder location.



Note: This is only an example, if FWTools was installed into a different location/directory the file path will differ

Insert the bin file location into the makeGeotiff.m file in MATLAB. This will only need to be done once, i.e. when GenerateGeotiff.m (makeGeotiff.m) is used for the first time. Open MATLAB and ensure that makeGeotiff.m are loaded into the current directory. Double click on make GeoTiff and the .m file should open in the MATLAB Editor (See example below).



Paste the file path to the bin file (I.e. “C:\Program Files\FWTools2.4.6\bin”) to where the program requires to edit the file path so that the file path in the program points to the computers install GDAL. Ensure that the file path once entered is in single invert commas and the text colour changes to blue. See example below.

```
function [im  topLat LeftLon bottomLat rightLon ] =makeGeoTiff(pathname,filename,tile,topLat,LeftLon,bottomLat,rightLon)
    gdalpath = 'C:\Program Files\FWTools2.4.6\bin\'; % edit this to point at your install of GDAL
    tempfile = [pathname '\temp.tif'];
    imwrite(tile,tempfile);
     dos([ '"' gdalpath 'gdal_translate" ' tempfile ' ' filename ' -of GTiff  -a_srs wgs84 -a_ullr ' ...
        num2str(LeftLon) ' ' num2str(topLat) ' ' num2str(rightLon) ' ' num2str(bottomLat)])
    dos(['del ' tempfile]);
end
```

Note: After copying your file path into makeGeotiff.m, a backslash (\) may need to be placed after “bin”, as illustrated in the example above)

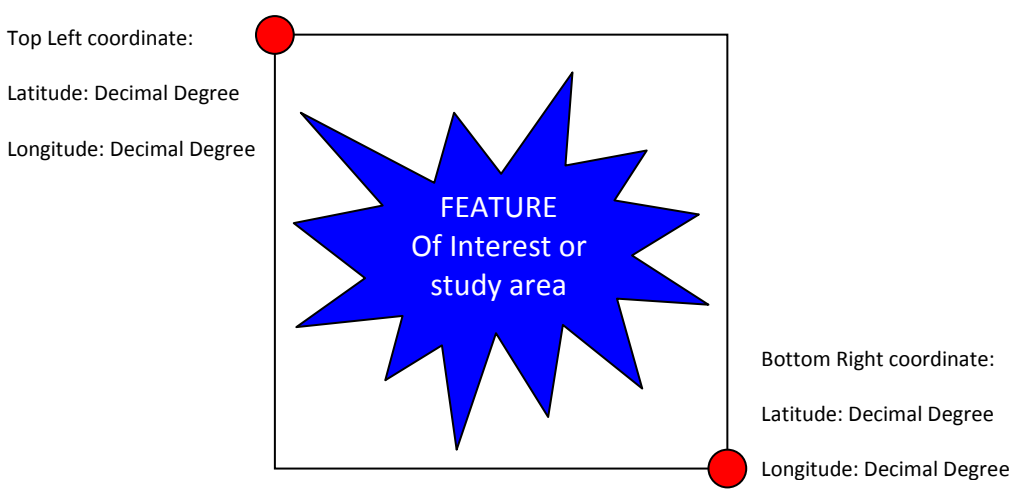
Close the editor and save the changes to makeGeotiff.m.

Step 4: Running GenerateGeotiff.m:

Data Acquisition:

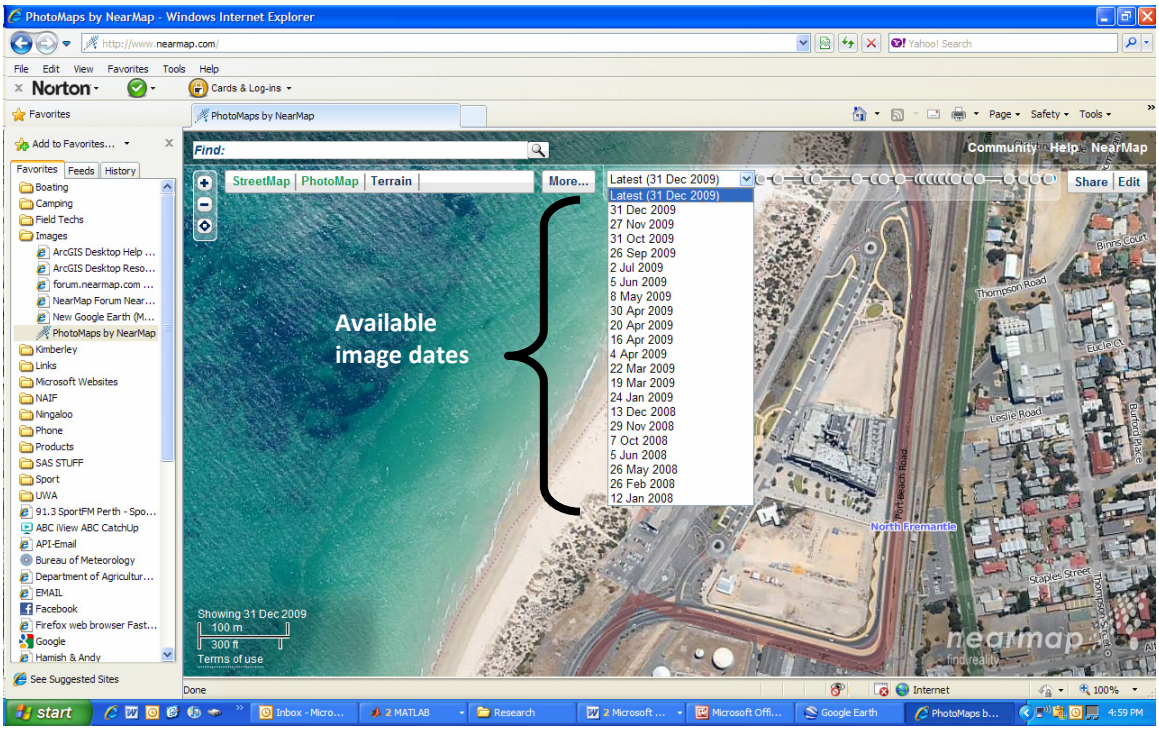
To Run GenerateGeotiff.m, there is some information which needs to be collected prior to the operation of the program.

The first piece of information required is the spatial location. To acquire the entire extent of the feature or study area one must imagine there is a square or rectangle box drawn around the feature or study area; as illustrated below. Find coordinates of the top left and bottom right corners of this imaginary box (red dots, below). These coordinates are what GenerateGeotiff.m (getTile.m) uses to attain the desired image. GenerateGeotiff.m (getTile.m and makeGeotiff.m) require the coordinates to be in WGS84 decimal degree format.



The top left and bottom right coordinates can be found either on the ground using GPS coordinates or identified from areal imagery. Google Earth is quite useful for this as it provides a latitude and longitude output for the cursor location. The settings in google earth can be changed to the required WGS84 decimal degrees format.

As GenerateGeotiffv1.m is currently setup for use with the Nearmap, which means an image date or datestring needs to be defined. Nearmap has series of images taken on different dates, therefore the date of the image required needs to be defined. In order to define the date check the available image dates on the Nearmap website (nearmap.com). I.e. The image below shows all available dates for imagery from Leighton beach Western Australia.



In the image above the latest date is the 31st of December 2009. The date string recognised by the Nearmap server is in the format of yyyymmdd. This means that when the date string is entered into GenerateGeotiff.m it must be in this format. I.e. for the latest imagery available in the example above the date string would read 20091231.

Data Input to GenerateGeotiff.m:

In MATLAB double click on the GenerateGeotiff.m in the MATLAB “current directory” to bring up the .m file in the editor.

a) Set the server:

Once this is opened the first piece of information which requires checking is the server from which the image is being attained. The server name must be in single Inverted commas. I.e. server= ‘nearmap’ (Note: That this is a string field).

b) Set the image coordinates:

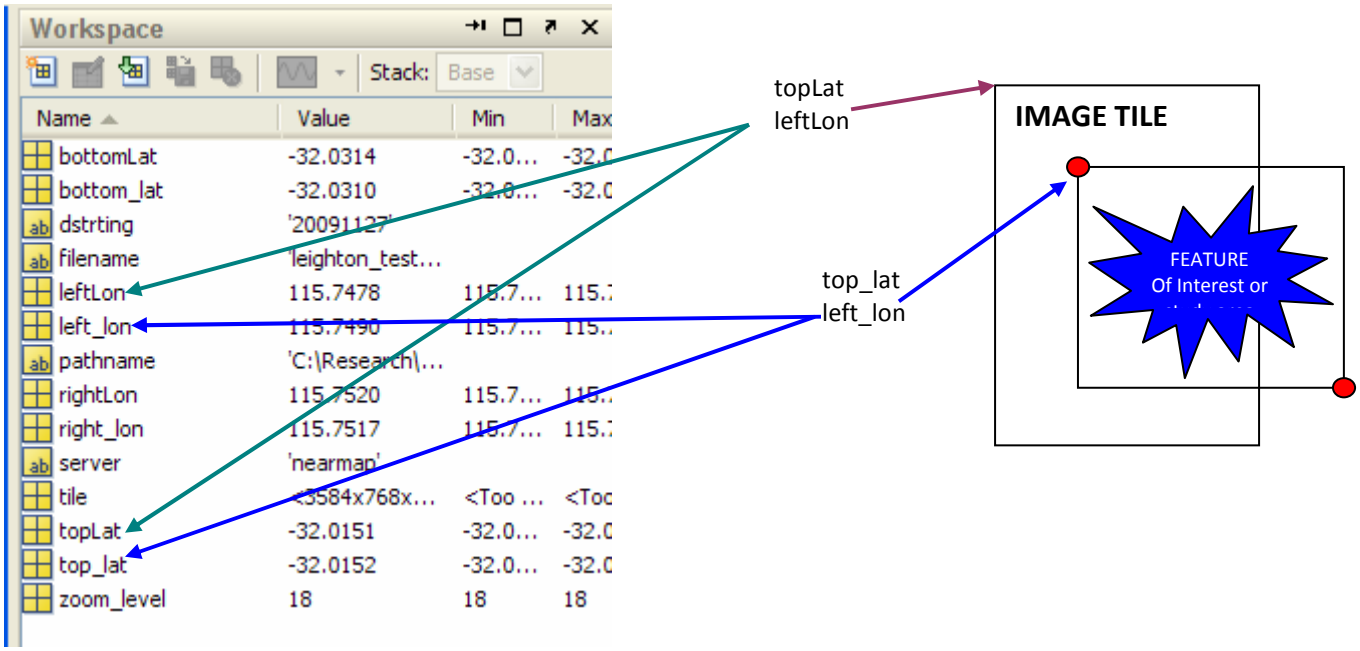
The toplat/leftlon/bottomlat/rightlon are the next parameters which are entered into GenerateGeotiff.m. There may already be numbers present for toplat/leftlon/bottomlat/rightlon, if so just replace these with the coordinates for the image that is required. The coordinates must be in WGS84 decimal degrees for the program to work accurately. Ensure that the semicolon is placed after the number (This ensures that the MATLAB output in the “command window” is suppressed). See below.

```
% Feature coordinates:

%These are the coordinates which bound the feature
%of interest from the top left of the feature to the bottom right. These
%coordinates should be in WGS84 Decimal Degrees. Note: Lat and Lon stand
%for both Latitude and Longitude respectivly.
top_lat=-32.015173;
left_lon=115.749022;
bottom_lat=-32.031041;
right_lon=115.751729;
```

Replace these latitudes and longitudes with latitudes and longitudes required to attain the desired image.

Note:
Once the script has run, there will be two entries for each latitude and longitude in MATLAB's workspace. The entry with the underscore will be the feature coordinates, where as the output without the underscore will be the image or tile coordinates. This occurs as the corner of a feature may be located in the middle of a tile. getTile.m can not take portions of tiles, hence has to take the entirety of the tile from where the corner of the feature is (See Below). Therefore the accurate coordinates for the extents of the image are topLat, leftLon, botLat, rightLon



c) Set the zoom:

Zoom is the zoom level or quality of the image being attained from the server. Nearmap and Google earth work on the same zoom levels, however Nearmap has the capacity to collect images at maximum zoom level of 21, while the maximum zoom of Google earth 18. Typically the higher the zoom level the higher resolution and larger the image file. Pending computational power and use of the image will depend on the zoom level from which the image is extracted at. I.e. Zoom = 18.

d) Setting the image Date:

The date string needs to be defined when images are being attained from the Nearmap servers. In section 4a the datestring(s) of potential images were identified. This datestring, I.e. 'yyyymmdd' can be entered into GenerateGeotiff.m as illustrated below. Ensure that the date string is placed between single inverted commas and the dte string is terminated with a semicolon.

Enter the datestring here

```
%Date of the image:

%For Nearmap.com images there is a large temporal
%resolution of images. This vairable identifies the time period from which
%to attain the image from. The date string should be entered between '' in
%a year, month, day format simular to, '20091127'. This indicates that you
%are attaining an image taken on the 27th of November 2009.

dstrting='20091127';
```

e) Enter Pathname:

The pathname is the directory or path where GeoTiff will be placed. This path can be anywhere on the computer provided that the pathname contains no spaces in any of the file and folder names. For ease of use, the pathname can be the same path as the current directory (I.e. something like; 'C:\Research\Matlab'). The Pathname must be placed between inverted commas and is terminated with a semicolon as illustrated in the example below.

```
%Pathname:

%This is path or folder where the generated Geotiff will be saved. The pat
%must be placed between ''.
pathname='C:\Research\Matlab';
```

f) Enter Filename:

The filename is what the Geotiff image will be called. The Filename must be placed between inverted commas and have .tif after the filename. The filename must be terminated with a semicolon as illustrated below.

```
%Filename:
%File name is the desired filename for the generated geotiff. The filename
%must be placed between '' and contain ".tif".
filename='leighton_test.tif';
```

If a MATLAB memory error is present in the “command window” after running this script, the imagery which is trying to be attained is likely to be in excess of 600MB, which is too large for MATLAB to process. If this is the case the image size should be reduced (I.e. Reduce the size of the Imaginary box in step 4 “Data Acquisition”) to less than 600MB allowing MATLAB to process the image.

If the Feature is larger than the attainable image, then several overlapping images may need to be attained in order to cover the geographical extent of the feature. These images can then be placed into a geographical or spatial mapping package and viewed simultaneously. Another way around this is use the makegrid.m MATLAB scripts if available.

From here the image can be removed or copied from its location and then used in geospatial programs. The default coordinates of the image when imported into the geospatial program will be WGS84 decimal degree format.

Section 2: Using makegridv1

Step 1: Setting up your computer to use makegrid.m:

The setting up of the computer is the same for makegrid.m as it is for GenerateGeotiff.m. If your computer is has not been setup for GenerateGeotiff.m, then please follow the steps on pages 3-10 to ensure that MATLAB and GDAL tools are installed and setup correctly.

To run makeGrid.m the files contained in the zipfile makegridv1.zip will need to be extracted into a normal folder. Once the files are extracted they can either be copied into the MATLAB current directory folder or the MATLAB current directory can be set to the location of the extracted files. See makeGeotiff, section 1 for more specific instructions and information. It would be advised that a copy of the initial zip folder is retained in case of corruption of the extracted files.

Once the files are copied from the zipfile or the current directory in MATLAB is set to the location of the extracted .m files the files required to run makegridv1.m should be loaded in the current directory. There should be a total of 17 m files in the current directory (See Appendix 1 for specific m files).

Step 2: Setting the Path to the Bin File:

See section 1, step 3. This outlines the specific steps how to find the location of the Bin file and then how to copy the location of the file path, and how to set this in MATLAB.

Follow the same steps to setup up the GDAL path for makegrid.m, however the .m file where the GDAL path will need to be changes is makeGeoTiff.m.

The GDAL path in the makeGeotiff.m script will need to be checked (and if not correct changed) regardless if your computer has been set up and runs generateGeotiff.m successfully.

Step 3: Running makegrid.m:

Data Acquisition:

See Section 1 step 4 (pp. 12-13) on how to get the correct corner points in order to attain a successful image of a feature.

Allowable Spatial extents in MATLAB:

MATLAB is restricted in the size and extent of arrays in which it can process. MATLAB can process any single array <600MB or a series of arrays which are <600MB and their total does not exceed 1500MB. This processing ability limits the spatial extent of which the images can be attained from the image server. As the computational size of an image is highly dependent on the zoom of the image, it is the zoom level (not xblocks or yblocks) which determines the spatial extent of the image.

For zoom level 18 MATLAB can handle images depicting <12 Km^2, where as for zoom level 21 MATLAB can only handle images <2 Km^2. This should be your guide when working out whether the feature of interest requires the attainment of several images continuously to cover the extent of the feature of interest.

If the feature of interest is larger than 2 Km^2 (12 Km^2) at zoom level 21 (18) then the image can be attained by running the script twice and changing the top and bottom latitudes or the left and right longitudes depending on the shape of feature.

Take this example; There is an area ~3.5 Km^2 which extends from latitude -31.99 to -32.04 and longitude 115.74 to 115.75. This is a tall skinny rectangle, therefore it makes sense to divide it in half somewhere along the Y axis (long edge). This means that the longitudes can be entered as normal, and these will not change during the two image attainments. The latitudes however will have to be changed. Enter the top latitude as normal in the script. However enter the bottom latitude as a point halfway between the top and bottom latitudes, I.e. 32.015 and attain the first part of the image. To attain the second part of the image, place this halfway latitude as the top lat in the script and enter the bottom latitude as normal and attain that image. Once both steps are completed all tiles which cover the extent of the feature should have been attained, with there being no gaps between any tiles.

Data input into makegrid.m:

In MATLAB double click on the makegrid.m in the MATLAB “current directory” to bring up the .m file in the editor.

a) Set the server:

Once this is opened the first piece of information which requires checking is the server from which the image is being attained. The server name must be in single Inverted commas. I.e. server= ‘nearmap’ (Note: That this is a string field).

b) Set the image coordinates:

The toplat/leftlon/bottomlat/rightlon are the next parameters which are entered into GenerateGeotiff.m. There may already be numbers present for toplat/leftlon/bottomlat/rightlon, if so just replace these with the coordinates for the image that is required. The coordinates must be in WGS84 decimal degrees for the program to work accurately. Ensure that the semicolon is placed after the number (This ensures that the MATLAB output in the "command window" is suppressed). See below.

```
% Feature coordinates:

%These are the coordinates which bound the feature
%of interest from the top left of the feature to the bottom right. These
%coordinates should be in WGS84 Decimal Degrees. Note: Lat and Lon stand
%for both Latitude and Longitude respectivly.
top_lat=-32.015173;
left_lon=115.749022;
bottom_lat=-32.031041;
right_lon=115.751729;
```

Replace these latitudes and longitudes with latitudes and longitudes required to attain the desired image.

See table 1 and table 2 in Appendix 2 for further information regarding the spatial extents of images, and image allotments.

c) Set the zoom:

Zoom is the zoom level or quality of the image being attained from the server. Nearmap and Google earth work on the same zoom levels, however Nearmap has the capacity to collect images at maximum zoom level of 21, while the maximum zoom of Google earth 18. Typically the higher the zoom level the higher resolution and larger the image file. Pending computational power and use of the image will depend on the zoom level from which the image is extracted at. I.e. Zoom = 18.

d) Setting the image Date:

The date string needs to be defined when images are being attained from the Nearmap servers. In section 4a the datestring(s) of potential images were identified. This datestring, I.e. 'yyyymmdd' can be entered into GenerateGeotiff.m as illustrated below. Ensure that the date string is placed between single inverted commas and the dte string is terminated with a semicolon.

Enter the datestring here

```
%Date of the image:

%For Nearmap.com images there is a large temporal
%resolution of images. This vairable identifies the time period from which
%to attain the image from. The date string should be entered between '' in
%a year, month, day format simular to, '20091127'. This indicates that you
%are attaining an image taken on the 27th of November 2009.

dstrting='20091127';
```

e) Enter Pathname:

The pathname is the directory or path where GeoTiff will be placed. This path can be anywhere on the computer provided that the pathname contains no spaces in any of the file and folder names. For ease of use, the pathname can be the same path as the current directory (I.e. something like; 'C:\Research\Matlab'). The Pathname must be placed between inverted commas and is terminated with a semicolon as illustrated in the example below.

```
%Pathname:

%This is path or folder where the generated Geotiff will be saved. The pat
%must be placed between ''.
pathname='C:\Research\Matlab';
```

f) Enter Filename:

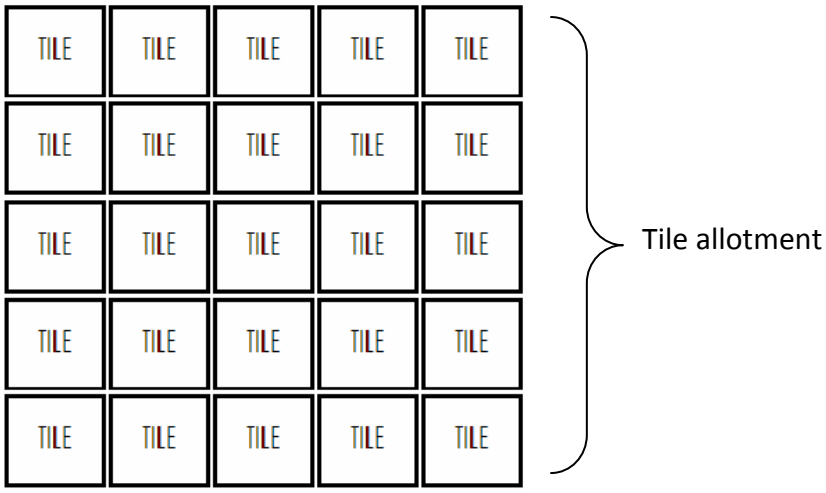
The filename is what the set of tile allotments will be called. The script will automatically assign the column and row information from where the tile allotment came after the filename. The filename format is “filename_column_row. I.e. Leighton_5_6, indicates that this image has a filename of “Leighton” and the allotment of tiles comes from column “5” row “6”. The Filename must be placed between inverted commas. The filename must be terminated with a semicolon as illustrated below.

```
%Filename:
%File name is the desired filename for the generated geotiff. The filename
%must be placed between ''.
filename='sect2';
```

g) xblock and yblock:

The xblock and yblock is how the user defines the groupings or allotments of tiles in which they desire. In makegrid.m the user can determine the size (both geographically and computer storage) of the tile allotments which it divides the image up in to. When setting the xblock and yblock the user is setting how many tiles in the x and y direction the user would like to divide the image up into.

For example if a user defined xblock as 5 and yblock as 5 then the each tile allotment will contain 5 tiles in the x direction and 5 tiles in the y direction. This totals 25 tiles in each allotment of tiles (I.e. 5*5=25). See below for Graphical example:



It is suggested that xblock and yblock are set the same, resulting in square tile allotments. Common xblock and yblock settings are either 5 or 10; As these work well with both low and high resolution imagery and are easily managed in geospatial programs while providing sufficient coverage area.

See Table 3 and 4 in Appendix 2 for how many meters are covered in the x and y direction (assuming a square) with each image allotment, and the computational size of each tile respectively.

makegrid.m can now be run, by either selecting Debug> Save and Run makegridv1.m or by pressing F5 or running a saved makegrid program in the command window. The script will now place the tile allotments into the folder which is defined by the path. From that folder the images can be copied or imported into geospatial programs or GIS systems.

.

Appendices:

Appendix1:

.m files which should be contained within makegridv1 zip file.

- GoogleTile.m
- LatLonToMeters.m
- LatLonToPixels.m
- MetersToTile.m
- MetersToPixels.m
- Resolution.m
- PixelsToTile.m
- TileBounds.m
- PixelsToMeters.m
- TileLatLonBounds.m
- MetersToLatLon.m
- PlotTileSize.m
- calcTile.m
- makeGeotiffGrid.m
- getImageFromTiles.m
- makeGeotiff.m
- makegridv1.m

Appendix2:

Table 1: The maximum extent of array which can be processed in MATLAB at 1 time. The extent is in Km^2

xblock ybolock		
Zoom Level	5x5	10x10
18	134	135
21	2	2

Table 2: The extent of each tile allotment in meters squared (m^2).

xblock ybolock		
Zoom Level	5x5	10x10
18	0.4	1.69
21	6.40E+-03	2.59E+-02

Table 3 illustrates how many meters in the x and y direction (assuming a square) are covered by each image allotment

xblock ybolock		
Zoom Level	5x5	10x10
18	650	1300
21	80	161

Table 4 the computational size of each image allotment in megabytes (MB).

xblock ybolock		
Zoom Level	5x5	10x10
18	4.71	18.70
21	4.71	18.70