DTM Vancouver To-do

**Misc**

1. ~~Download Vancouver from firebug and see if it is ok.~~
2. ~~Add a command to import dtm, and try to import the large data set. See how MS handles it~~
3. ~~Clean up the policy files.~~
4. ~~Put the right file header on all files~~

*/\*--------------------------------------------------------------------------------------+  
|  
| $Source: mstn/mscore/moss/LineStyle.cpp $  
|  
| $Copyright: (c) 2012 Bentley Systems, Incorporated. All rights reserved. $  
|  
+--------------------------------------------------------------------------------------\*/*

**Core**

1. ~~Change Core to use T\_Super~~
2. ~~Remove all .fdf files. I only care about the 2d fns header file. and maybe not even that one,~~
3. ~~Implement custom extended data attachment to DTM~~
   1. ~~Change existing code to use the new attachment.~~
   2. Need Extract and Replace extended functions.
   3. Need register AppData functions?
4. ~~Change some of the names of classes, DTM\_USER\_TAG etc to DTMUserTag/DTMFeatureId~~
5. ~~Change to use correct Int64 instead of \_\_int64~~
6. ~~Move Public Enums out to new file (moved some to TerrainModel.h)~~
7. ~~Also move the DTMFeatureID and DTMUserTag.~~
8. ~~Implement transform interface.~~
9. ~~Change make functions to Create.~~
10. ~~Publish the TerrainElement header files~~
11. ~~Add enum for DTMState and DTMFeatureState and DTMRollback.~~
12. ~~Change DTMRollback to DTMCleanupOptions and add the clean up enum to functions.~~
13. ~~Move the xm functions from dtm2dfns to dtm2dfnsxm~~
14. Create header for TM structures and TM enums.
15. ~~Write quick exe to remove duplicate declarations. + find out if they are used (They might not have any implementation) Also remove multiple blank lines.~~
16. Move all into the Bentley::TerrainModel namespace, rename the Bentley::DTM aswell.
17. ~~Remove BC\_NOMDLENV stuff.~~
18. ~~Do the managed API.~~
19. Change the unmanaged API to more match the managed one. Hopefully the managed API is just wrapper calls.
20. ~~Create a Unit Test.~~
21. ~~Remove aliases for feature type just use one~~
22. Look at managed TerrainModel API.
23. Change functions to return DTMStatusInt.
24. Clean up BcDTM interface to match the managed API. (have 1 to 1 functions so that the managed are just simple methods calling the unmanaged.)
25. ~~Look at code using -98989898 and -87878787 should be using the #def~~.
26. ~~Rename the inline cpp file to a header file in PrivateAPI.~~
27. Look at implementing the new Triangle Graphic Break feature.
    1. Check if p123 is a triangle if so delete the feature.
    2. Try and make P123 into a triangle by swapping sides
    3. If succeeded store feature in rollback.
    4. Otherwise delete the feature.
28. ~~Change NATDLLEXP to a more verbose version BENTLEYTMCORE\_DLLEXPORT~~
29. Implement error statuses.
30. Move the browseSourceFeatures into core, and create a browseFeatures using the same method.
    1. Make sure it copes with the current feature set.
    2. Remove the old feature browsing functions.
    3. Do the same for browsing error features.
    4. Duplicate Points?
    5. Crossing Features.
31. ~~Change to new and delete for BC\_DTM\_OBJ and move some of the initialization to the constructors and destructors.~~
32. ~~Create a small struct which can be passed into the contour functions~~.
33. ~~Sort out declspec on exported functions at the moment they aren’t set on some of the class functions.~~
    1. ~~Sort out the #define for DllExports if CREATE\_STATIC is defined then always no dllimport/export~~
    2. ~~Rename the formatting one.~~
34. Change to use more std::vectors instead of \* and num.
35. ~~Remove CVSID lines.~~
36. ~~Remove transformation wrapper class and move into BcDTM class.~~
37. Merge changes from Civil Ext into DTM.
38. Look at implementing a Directional Hull Line Feature. (eg to the left.)
39. Look at polygon cleaning
40. Look at merge when they don’t overlap, need to just add the terrain with voids.
41. Clean up the header files so that they can be included on their own.
42. Put the correct case on API functions.
43. Change some Enums values to be better defined.
    1. DTMFenceType
    2. DTMFenceOption
    3. DTMFeatureType
    4. DTMCleanup
    5. DTMState

**Formats**

1. ~~Move MX/Inroads/ERSI import into formats.~~
2. ~~Look at an import/export provider for formats. (To allow plugins? In the end)~~
3. Convert the LandXML Export implementation.
4. Create DEM Import
5. Move MX Modelfile reader into formats.
6. Look at moving DTMLink into formats.
7. Do the managed API.
   1. ~~LandXML import.~~
   2. LandXML export.
   3. Inroads.
   4. ~~MX.~~
   5. Lidar.
8. Clean up the header files so that they can be included on their own.
9. Compare output/input from landXML imports/exports

**Drainage**

1. Create a DrawContoursWithDepressions in the Drainage dll and remove depressions stuff from Core.
2. Move/or Remove the Drainage stuff into the Drainage dll.
3. Clean up the header files so that they can be included on their own.

**Element**

1. ~~Look at the element template code.~~
2. Look at some of the ToDo sections.
3. ~~As the ref doesn't change I can remove the code which keeps these in sync.~~
4. ~~Move the text caching into own file.~~
5. Changed to use Foreach (remove the for each we have and use the proper one)
6. ~~Change Core to use T\_Super~~
7. ~~Create an interface to allow caching to work. (Look at T\_HOST.GetGraphicsAdmin()) <--~~ Need ~~the view handler this may be interesting. not sure what happening re view index!!!~~
8. ~~Add appdata to the dgnModel to handle the SetDgnCacheUnloadingFunction unload mechanism.~~
9. Need to get the MstnApp to set the Active DGNModelRef possibly handle the name.
10. ~~Send email to see how to handle the sub display types folders etc.~~
11. ~~Look at managed DGNPlatform element API~~
12. Add all methods called from ElementHandler in the class API. Eg not use the fns methods.
13. Clean up the header files so that they can be included on their own.
14. Can we implement the progressive display stuff for displaying the DTM triangle tiles.
15. ~~Don’t bother with depressions if the symbology is the same for both.~~
16. ~~Sort out PointDisplayParams~~
17. ~~Sort out drawing of points move into a single class and simplify and use IDisplaySymbol to improve speed.~~
18. ~~Improve tile caching to get proper extents do we draw triangles which span multiple tiles.~~
    1. ~~Write function to get the new extents from the range~~
19. Add functions to convert slope and aspect.
20. Add Pick ElemTopology for points, so we can show tooltip of the information.
21. ~~Put some LOD in for symbols if they are too small to see in the display. If so then just display points.~~
22. ~~Change MstnPlatform to include the TerrainModel stuff.~~
23. ~~Improve the handling of Caching classes, use the classes to remove from the list, and use the AppDatas for DgnCache and DgnModel, possibly DgnFile. to handle modifications.~~
24. ~~Fix Material ID This is 64 bit and not 32 bit.~~
25. ~~Create function to get the right extents of the tile.~~
26. If you could move to using the MaterialId::Serialize & MaterialId::Deserialize methods for vancouver for persisting references to materials in the file that would be the best solution. Due to legacy reasons we still support .pal material files which are external non dgn files and can only be referenced by the material name.
27. You should also look at implementing the IMaterialPropertiesExtension for your element. Then the Microstation material editor will be able to assign materials to your elements. Also if you have sub components to your elements i.e. faces on smart solids it can assign materials to those too. Using this extension is how DgnPlatform will look for materials on your elements in future.
28. ~~Check that the Vancouver branch is all added then move the MERGE\_platform081109xx\_hg to the tip.~~
29. On Clean up temporary changes change to use a template function or a normal function.
30. ~~Implement Symbology Override~~
31. Fix clipping, using an element is ok but using selected clip isn’t
32. ~~Make DTMDataRef always stay around as the ElementRef and XAttribtues wont move.~~
33. Look at the DTMDataRef caching as this can be improved and simplified.
    1. ~~We can store a BcDTMObject in DTMDataRef, and update the data in this, this will make Rob happy as they will be 1 DTM instance and we can put this in the btree as well.~~
    2. ~~Make bcDTM look at the btree for DTMElements.~~
    3. ~~Store the stuff and caching etc in ElementRef AppData~~
    4. Move all XAttribute DTM stuff into DTMXAttributeHandler.cpp
34. Create a precompiled header for handlerApp.
35. ~~Move rest of the DisplayHandlers into handlerApp.~~
36. ~~Move the TerrainModel handler into DgnPlatform~~
37. ~~Move the TerrainModelApp into DgnDisplay~~
38. ~~Can we move out the DTM DataRefs for both MrDTM and DTMElementDataRef~~
39. ~~Store the Hull in XAttributes and use this to display.~~
    1. ~~Implement the CreateXGraphics Function which is in the 106 element.~~
40. Simplify the display handlers as most might be able to use a template, or base class.
41. ~~Move out DTMElementHandlerManager into DisplayHandler~~
42. ~~Rename ElementHandler to TerrainModelDgnPlatform and DisplayHandler to ElementHandler.~~
43. ~~When drawing the Hull when nothing else is drawn use the symbology of the Hull sub display.~~
44. ~~Create a list of DTMDisplayCaches so that DeleteCacheFromView can work.~~
45. Change delete delay tiles, so that when we remove tiles put in a reference list, and clear this list out. Change refcounting to use RefCountedPtrs
46. ~~Move the XAttribute Handling from DTMDataRef into DTMXAttributeHandler.~~

**Commands**

1. ~~Implement the terrainmodel commands.~~

Full implementation.

1. ~~Small handler inside DgnPlatform, only draws the hull (saved as XGraphics). (doesn’t have the sub handlers registered?)~~
2. ~~DgnDisplay has the handlers as they are now. (registers the sub handlers?)~~
3. ~~PowerPlatform has some more extensions.~~

**Major bits todo**

1. ~~Element Info (Icons/override symbology/ByCell editors)~~
   1. ~~Add ByCell(FromParent) Editors~~
   2. ~~Override Symbology stuff~~
   3. ~~Commit the RuleSet and fix file ruleset~~
   4. ~~Add relationship to get the displayparams.~~
2. ~~ElementNet API~~
3. ~~Element Templates~~
   1. ~~Add Templates~~
   2. ~~Implement Apply symbology~~
4. TM commands.
   1. Annotate Contours
      1. ~~Get command working~~
      2. Fix Dialog.
   2. Annotate Spots
      1. ~~Get command working~~
      2. Fix Dialog.
   3. ~~Add icons~~
   4. ~~Add Tasks and fix menu.~~
   5. ~~Localization~~
5. ~~Implement DisplayStyle Caching~~
6. ~~Remove transformation wrapper class and move into BcDTM class.~~
7. ~~Move the TerrainModel handler into DgnPlatform.~~
8. ~~Move the TerrainModelApp into DgnDisplay~~
9. Sort out drawing Drainage stuff/depressions/low/high points.
10. ~~Localization.~~
11. ~~Put the correct case on API calls.~~