DataStatus API

DATA EXTENTS:
Float* getLocalExtents()
DataMgr could provide. With multiple grids need per-variable and overall
Float* getFullSizes()
DataMgr could provide.
Float* getFullStretchedSizes()
DataMgr could provide.
float getVoxelSize()
DataMgr could provide

Float* getStretchedExtents()
Float* getStretchedLocalExtents()

Response: A DataMgr::GetVaribleExtents() method exists, which can be used to query extents of any variable at any time step. The method will cache results based on variable coordinates: two queries to the same variable, both using the same coordinate variables, will result in a cache hit on the second query.

The DataMgr::GetVaribleExtents() supports the functionality of the DataStatus::getFullSizes() methods. The other, above DataStatus methods will not be part of the DataMgr.

Note: GetVariableExtents() is potentially an expensive method (if results aren't cached), and should only be called on-demand.

TIMESTEPS
DataMgr could provide these:
getMinTimestep() [actually min for which there is data]
getMaxTimestep()
getNumTimesteps()

Response: the current DataMgr::VariableExists() method will be overloaded as follows to support the getMinTimestep() and getMaxTimestep() functions:

VariableExists(ts, varname, lod, level)
VariableExists(ts) - returns true if **any** data is present at time step
VariableExists(varname) – returns true if indicated variable is present at **any** time step

John Clyne 9/12/2014 2:14 PM
Formatted: Font:Bold
John Clyne 9/12/2014 2:14 PM
Formatted: Font:Bold

The DataMgr:GetTimeCoordinates() returns a vector of available time coordinates. The vector length is the number of time steps.

DATA PRESENCE:
DataMgr could provide all of these:
Bool dataIsPresent2D(timestep)
dataIsPresent3D(timestep)
dataIsPresent2D()
dataIsPresent3D()
dataisPresent2D(varname, timestep)
dataIsPresent3D(varname,timestep)
dataisPresent2D(varname)
dataIsPresent3D(varname)

Response: See discussion on VariableExtents() above.

int maxXFormPresent3D (varname, ts)
int maxLODPresent3D(varname, ts)

Response: See discussion on VariableExtents() above.

bool fieldDataOK(refinement, lod, timestep, U,V,W)

Response: The DataMgr will not provide this.

float getDefaultDataMin(varname)
float getDefaultDataMax(varname)
void setDataMissing3D()

Response: The DatMgr will provide a GetDataRange(ts, varname,lod,level) function to return the min and max values of a variable at a given time step. If the results aren't available in cache this method will need to read the data from disk, etc.

DATA MAX/MIN
DataMgr could provide:
getDataMax3D(varname,timestep, bool mustGet)
getDataMin3D(varname,timestep, bool mustGet)

Response: The DataMgr will not provide these.

METADATA etc:
DataMgr already provides. Will they change with 3.0?
getNumTransforms()

getNumLODs()

Response: These exist on the DataMgr as: GetNumRefLevels() and GetCRatios()

getProjectionString()
getVDCType()
getCacheMBs()

Response: The above will be added.