

VTK 4.0

Architectural Changes

1. Require ANSI compiler - STL implementation allowed
2. Cmake – an improved build environment
3. Wrapping technology – wrap all methods, dispense with //BTX //ETX, etc.
4. DataSetAttributes represented with FieldData
(Part 1: GetScalar() returns a vtkScalar*, and things remain compatible;
Part 2: GetScalar() returns a vtkDataArray*, change GetScalar() to GetTuple(); this breaks every filter)
5. File Formats – support efficient seeks; reading parts; other information like bounding box and scalar range.
6. Remove old-style callbacks in favor of Observer/Command
7. Templating
 - Classes (Collection, DataArray)
 - Functions (filters, etc.) – performance improvements in filters
8. Add Exceptions
9. Image/Render Window - coalesce the classes; added style/interaction to imaging
10. Const correctness – evil but correct, will break user code
11. Traits/enums for class constants – rather than #define
12. Reorganize directory structure
 - Source directories: core, common, imagingFilters, graphicsFilters
 - Examples out of source tree
 - Vtkbaseline – don't grab images you don't need
13. Testing support (XML, other?)
14. Support for libTiff, png, zlib. Jpeg
 - Integrated into VTK?
 - Or as object-factory addition
15. friend functions rather than //BTX and //ETX
16. Support time in the data pipeline
17. Rename Extent -> Region (ala Insight)
18. Namespace vtk
19. Support serialization (XML) of instances
20. Use of smart pointers (rather than explicit register/unregister reference counting)
21. OpenGL performance/issues
 - glContext shared
 - managing gl state
 - solve the mystery of multiple actors/performance hit

Added Functionality

1. Unstructured grid volume rendering
2. Incremental rendering
3. Remove marked legacy code
4. Texture matrices
5. Feature extraction, large data visualization, etc.