ParaQ level view classes that interface to GUI
* Inits CS wrapped vtkMantaLib
* forces use of SMMantaReps and PVMMantaView

ServerManager View classes to control the remote/parallel vtk views
PVRenderView

Render() {
1) updates data into reps
2) gathers info
3) sends RQST_PREP to reps to move renderable geom to renderwindow
4) sends RQST_REN to reps which do ordered compositing
5) calls SyncRW(BeginRender)
}

PVMantaView

substitutes MantaRenderer, Camera and Light for 3D vis Data layer
GeometryRepresentation, UnstructuredGridRepresentation, UniformGridRepresentation, MantaGeometryRepresentation
MantaRep uses MantaSurfaceRep for pts, lines, surface rendering

SMRepresentationProxy

SMPVRepresentationProxy  ➔  vtkPVCompositeRepresentation

---

top level representation for something shown in a view

it holds many internal representations (see next three pages)
each sub representation shows the thing in a different way (volume render/surface/outline etc)
this will show 1 subrepresentation from that set
and at the same time show the selection and cube axis sub representations

3 "class" hierarchies, SM classes, XML configurations of those, and VTK level
PV pipeline and display of one filter in it into one RenderView

blue boxes are SM wrappers that ParaView uses for control
black boxes are VTK classes possibly instantiated remotely and in parallel
selection internally has a
text representation for labels and
a geometry representation to highlight selected entities
SMRepresentationProxy

SMChartRepProxy -> vtkXYChartRep

XYChartRep, XYBarChartRep

SMRepresentationProxy

SMSpreadsheetRepProxy -> vtkSpreadsheetRep

SpreadsheetRepresentation
PVDataRepresentation

reps from VTK lib

Spreadsheets Rep
ChartRep
TextRep
ImageSlice Rep
LabelRep
GeomRep
ImageVol Rep
UGridVol Rep
Selection Rep
CubeAxis Rep

VTK level representations

PVComposite Representation

GeomRep WithFaces

PVMantaRep
sustitutes MantaMappers, Actors, Properties
DataRep

AddTo/RemoveFrom(View)
Select(View, sel) and friends
Annotate(View, sel) and friends
GetShallowCopyOfInput(port, connection)

internal pipelines use this to get a hold of a shallow copy of data to show do it dynamically so that modified times can prevent stale data

VTK Reps

Qt, Geo, Rendered
(Tree, Graph, IICoord, Surface)

PVDataRepPipeline

shunts upstream pipeline updates depending on state of cache (which is left up to subclass to implement/use)

VTK Selection

vtkSelection

links selections in many reps

VTK Annotation Links

vtkAnnotationLinks

links selections in many reps

PVDataRep

Visibility - if off View passes ignored
UseCache/CacheKey

MarkModified() PV sets this on all processes so RS and CLI with no input will execute internal pipelines
RUE() splits into rank/file
RD() produces NULL since PV reps are meant to be sinks
ProcessViewRequest()
After update PVRenderView gathers info and sets up pipeline for remote/local render, B2F distributor, use LOD path, etc. GF must execute to get geometry size?

If translucent, RenderServer nodes sort B2F in Distributors depending on geom size and settings geom is send to render server and/or client.

Synchs watch window events and propagate camera/view state from client to servers. In SS rendering, synchs also watch events to composite on server and then deliver pixels from server to client.
Parallel window synchronization

SynchronizedRenderWindows

// propagates render and events

Identifier
bool ParallelRendering
bool EventPropagation
static global id to syncrenderwindows map for RMI

HandleStartRender()
// calls render on all processes and then master sends rest window parameters before render occurs

MultiProcessController

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow

RenderWindow