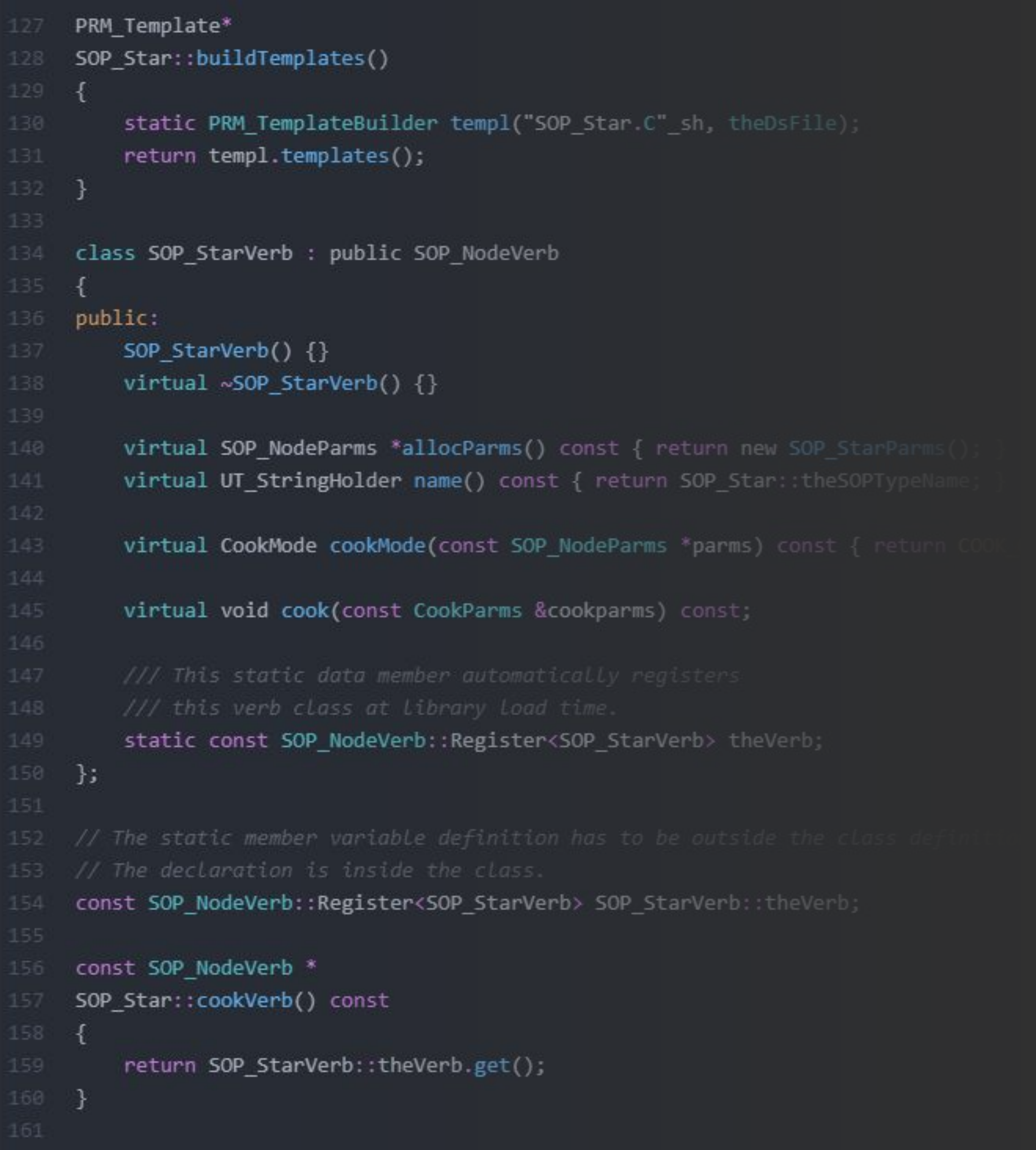


```cpp
PRM_Template*
SOP_Star::buildTemplates()
{
    static PRM_TemplateBuilder templ("SOP_Star.C"_sh, theDsFile);
    return templ.templates();
}

class SOP_StarVerb : public SOP_NodeVerb
{
public:
    SOP_StarVerb() {}
    virtual ~SOP_StarVerb() {}

    virtual SOP_NodeParms *allocParms() const { return new SOP_StarParms(); }
    virtual UT_StringHolder name() const { return SOP_Star::theSOPTypeName; }

    virtual CookMode cookMode(const SOP_NodeParms *parms) const { return COOK

    virtual void cook(const CookParms &cookparms) const;

    /// This static data member automatically registers
    /// this verb class at library load time.
    static const SOP_NodeVerb::Register<SOP_StarVerb> theVerb;
};

// The static member variable definition has to be outside the class defini
// The declaration is inside the class.
const SOP_NodeVerb::Register<SOP_StarVerb> SOP_StarVerb::theVerb;

const SOP_NodeVerb *
SOP_Star::cookVerb() const
{
    return SOP_StarVerb::theVerb.get();
}
```

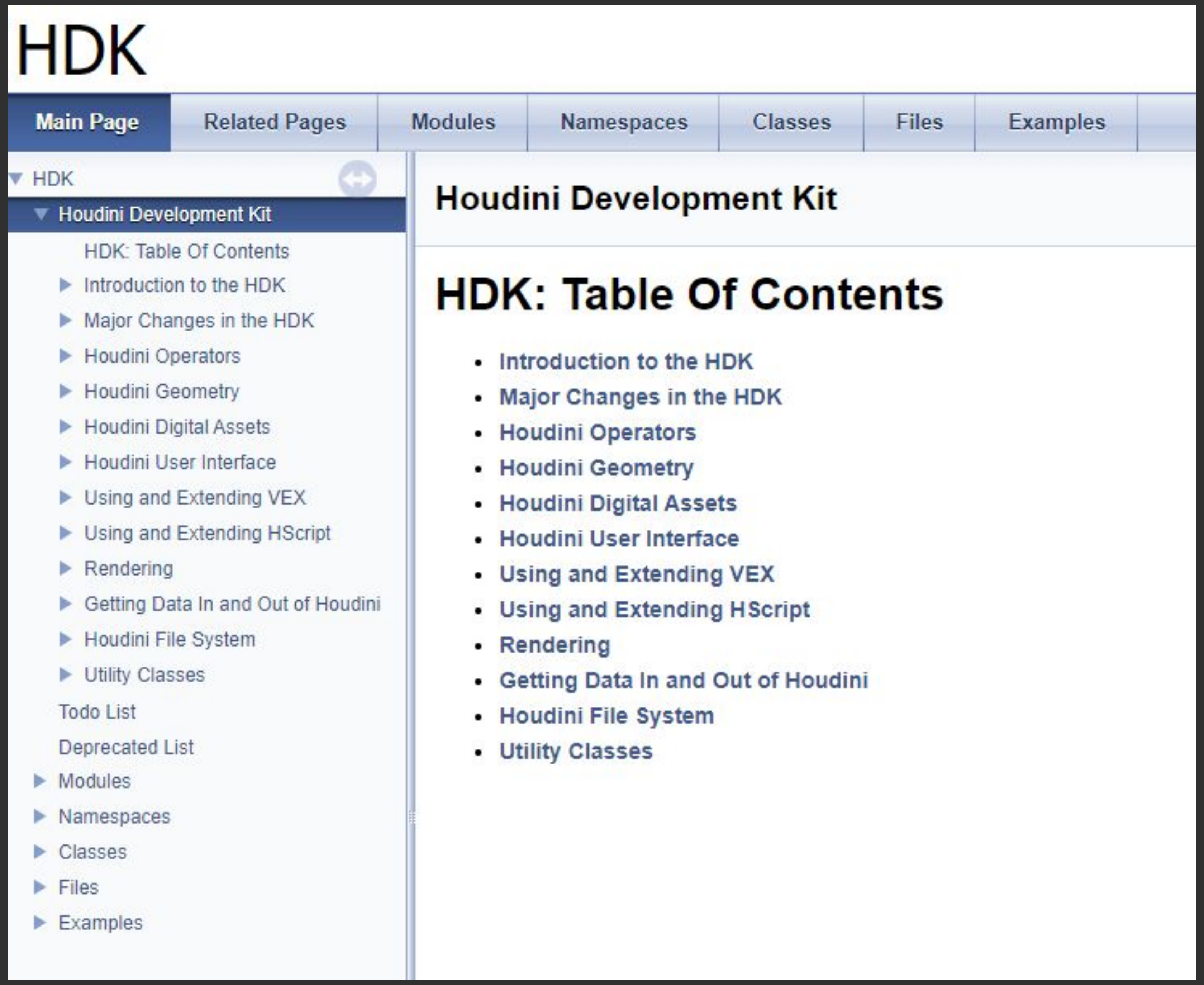
HDK

HDK for SOPs

Derrick Moser | SideFX

SideFX®

Where to find documentation?



- HDK
  - Houdini Development Kit
    - HDK: Table Of Contents
    - Introduction to the HDK
    - Major Changes in the HDK
    - Houdini Operators
    - Houdini Geometry
    - Houdini Digital Assets
    - Houdini User Interface
    - Using and Extending VEX
    - Using and Extending HScript
    - Rendering
    - Getting Data In and Out of Houdini
    - Houdini File System
    - Utility Classes
  - Todo List
  - Deprecated List
  - Modules
  - Namespaces
  - Classes
  - Files
  - Examples

Houdini Development Kit

HDK: Table Of Contents

- Introduction to the HDK
- Major Changes in the HDK
- Houdini Operators
- Houdini Geometry
- Houdini Digital Assets
- Houdini User Interface
- Using and Extending VEX
- Using and Extending HScript
- Rendering
- Getting Data In and Out of Houdini
- Houdini File System
- Utility Classes

https://www.sidefx.com/docs/hdk/

https://www.sidefx.com/docs/hdk

- overview
- class documentation
- required packages for development
- hcustom/generate_proto.py

`$HFS/toolkit`

- headers and example files

Houdini Development Kit
| sidefx.com/docs/hdk

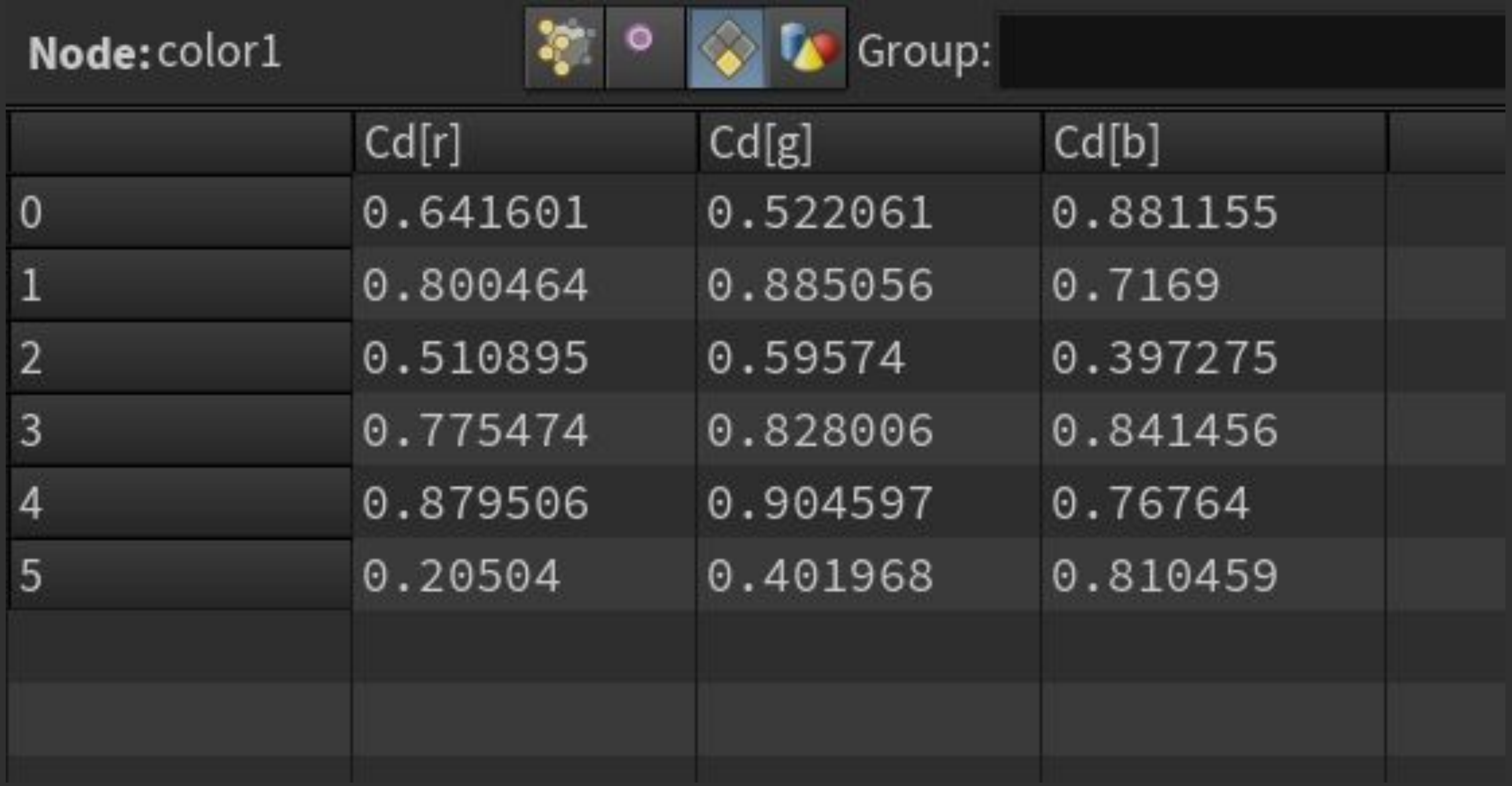
Geometry Detail Overview

container for geometry components

contains lists of: points, primitives, vertices

spreadsheet view

- rows: components
- columns: attributes, groups



Node: color1 Group:

	Cd[r]	Cd[g]	Cd[b]
0	0.641601	0.522061	0.881155
1	0.800464	0.885056	0.7169
2	0.510895	0.59574	0.397275
3	0.775474	0.828006	0.841456
4	0.879506	0.904597	0.76764
5	0.20504	0.401968	0.810459

Geometry Detail Overview

points

- a point is simply a position in space defined by a vector

primitives

- primitives are units of geometry above a point
- houdini supports several different types of primitives: eg. polygon, NURBS curve, metaball, etc.

vertices

- a vertex is a reference to a point
- primitives use vertices to reference points (eg. the corners of a polygon, the center of a sphere, or control vertices of a spline curve)
- primitives can share points, while vertices are unique to a primitive

GA *($HFS/toolkit/include/GA)*

- Geometry Attributes (base level of geometry classes)
- contains the base classes for all geometry in Houdini
- this library contains the definitions of many classes, though some are virtual abstractions implemented in higher libraries

GEO *($HFS/toolkit/include/GEO)*

- Geometry Library
- a 3D sub-class of the GA library
- this library defines most specialized classes for 3D data structures, including primitives
- primitive types

GU *($HFS/toolkit/include/GU)*

- Geo-Utility Library
- sub-classed off the GEO library

Higher level tools such as `GU_Detail::cube()`, `GU_Detail::polyIsoSurface()`, `GU_Detail::lsystem()` are implemented here.

Identifying Components

GA_Index

- enumeration
- GA_INVALID_INDEX

GA_Offset

- immutable location of component in internal arrays
- GA_DETAIL_OFFSET
- GA_INVALID_OFFSET

GA_IndexMap

- indexFromOffset()
- offsetFromIndex()

GA_AttributeOwner

- GA_ATTRIB_VERTEX
- GA_ATTRIB_POINT
- GA_ATTRIB_PRIMITIVE
- GA_ATTRIB_GLOBAL

Points

GA_Detail

- appendPoint()
- appendPointBlock()
- destroyPointOffset()
- destroyPoints()

- getNumPoints()
- getNumPointOffsets()
- pointOffset()
- pointIndex()
- getPos3()
- getPos3D()

Primitives

Creation

`GEO_PrimPoly::build()`

`GEO_PrimSphere::build()`

`GU_PrimNURBSurf::build()`

etc.

GA_Primitive

- `getMapOffset()`

GA_Detail

- `getPrimitive()`
- `appendPrimitivesAndVertices()`
- `getNumPrimitives()`
- `getNumPrimitiveOffsets()`
- `primitiveOffset()`
- `primitiveIndex()`
- `destroyPrimitiveOffset()`
- `destroyPrimitives()`

Vertices

GA_Primitive

- `getVertexCount()`
- `getVertexOffset()`
- `getPointOffset()`
- `setPointOffset()`

GA_Detail

- `vertexPoint()`
- `vertexPrimitive()`
- `pointVertex()`
- `vertexToNextVertex()`
- `getPrimitiveVertexCount()`
- `getPrimitiveVertexOffset()`
- `getTopology().wireVertexPoint()`

Example Creating Polygons

```
GEO_PrimPoly *poly =
     GEO_PrimPoly::build(gdp, /*nverts*/ 3,
                         /*open*/ false,
                         /*appendpts*/ false);
for(GA_Size i = 0; i < 3; ++i)
{
    GA_Offset pt = gdp->appendPoint();
    gdp->setPos3(pt, pos[i]);
    poly->setPointOffset(i, pt);
}
```

Example Creating Polygons

```
GA_Offset vtx;
GA_Offset offset =
        gdp->appendPrimitivesAndVertices(GA_PRIMPOLY,
                                         1, 3, vtx, true);
GA_Offset pt = gdp->appendPointBlock(3);
GA_Topology &topo = gdp->getTopology();
for(GA_Size i = 0; i < 3; ++i, ++pt, ++vtx)
{
    gdp->setPos3(pt, pos[i]);
    topo.wireVertexPoint(vtx, pt);
}
```

Iterating Over Components

GA_Range

rules for generating sequence of **`GA_Offset`**

```
GA_Detail::pointRange()
GA_Detail::vertexRange()
GA_Detail::primitiveRange()
```

GA_Iterator

```
- advance()/operator++()
- blockAdvance()
```

Example Iterating Over Points

```
for(GA_Iterator it(gdp->getPointRange(pt_group);
    !it.atEnd(); ++it)
{
    GA_Offset offset = *it;
    gdp->setPos3(offset,
                 plane.project(gdp->getPos3(offset)));
}
```

Example Iterating Over Points

```
GA_Offset start, end;
for(GA_Iterator it(gdp->getPointRange(pt_group));
    it.blockAdvance(start, end);)
{
    for(GA_Offset offset = start; offset < end; ++offset)
    {
        gdp->setPos3(offset,
                     plane.project(gdp->getPos3(offset)));
    }
}
```

Primitive Types: Polygon, VDB, Metaball, NURBS Curve, etc.

GEO_Primitive

- getTypeId()
- getTypeName()
- getLocalTransform()
- iterateEdges()
- enlargeBoundingBox()
- calcVolume()
- calcArea()
- calcPerimeter()

GEO_Detail

- countPrimitiveType()
- getGEOPrimitive()
- getPrimitiveTypeId()

Intrinsics

- findIntrinsic()
- getIntrinsic()
- setIntrinsic()

Example Checking Primitive Type

```
for(GA_Iterator it(gdp->getPrimitiveRange();
    !it.atEnd(); ++it)
{
    GA_Offset offset = *it;
    switch(gdp->getPrimitiveTypeId(offset))
    {
        case GA_PRIMPOLY: doPoly(offset); break;
        case GA_PRIMSPHERE: doSphere(offset); break;
        case GA_PRIMVDB: doVDB(offset); break;
        default: break;
    }
}
```

Example Reading Intrinsics

```
const GA_Primitive *prim = gdp->getPrimitive(prim_offset);

UT_String filename;
GA_LocalIntrinsic filename_h = prim->findIntrinsic("abcfilename");
if(filename_h != GA_INVALID_INTRINSIC_HANDLE)
    prim->getIntrinsic(filename_h, filename);

float frame = 0.0;
GA_LocalIntrinsic frame_h = prim->findIntrinsic("abcframe");
if(frame_h != GA_INVALID_INTRINSIC_HANDLE)
    prim->getIntrinsic(frame_h, frame);
```

GOP_Manager

Parse Groups

- `parsePrimitiveGroups()`
- `parsePointGroups()`
- `parseEdgeGroups()`
- `parseVertexGroups()`

Parse Detached Groups

- `parsePrimitiveDetached()`
- `parsePointDetached()`
- `parseEdgeDetached()`
- `parseVertexDetached()`

Example Parsing Groups

```
GOP_Manager gop;
const GA_PrimitiveGroup *group = nullptr;
if(group_pattern)
    group = gop.parsePrimitiveGroups(group_pattern,
                                     GOP_Manager::GroupCreator(gdp));
fpreal area = 0.0;
for (GA_Iterator it(gdp->getPrimitiveRange(group)); !it.atEnd(); ++it)
{
    area += gdp->getGEOPrimitive(*it)->calcArea();
}
```

Group Creation

Creation

- `newPointGroup()`
- `newVertexGroup()`
- `newPrimitiveGroup()`
- `newEdgeGroup()`

Internal

- `newInternalPointGroup()`
- `newInternalVertexGroup()`
- `newInternalPrimitiveGroup()`
- `newInternalEdgeGroup()`

Detached

- `createDetachedPointGroup()`
- `createDetachedVertexGroup()`
- `createDetachedPrimitiveGroup()`
- `createDetachedEdgeGroup()`

Groups

GA_Detail

- findPointGroup()
- findVertexGroup()
- findPrimitiveGroup()
- findEdgeGroup()
- destroyGroup()
- pointGroups()
- primitiveGroups()
- vertexGroups()
- edgeGroups()

GA_GroupTable

- begin()
- end()

GA_Group

- getName()
- isInternal()
- clear()
- entries()

Example Primitive Group

```
GA_PrimitiveGroup *bad_prims =
            gdp->newPrimitiveGroup("degenerate_prims");
for(GA_Iterator it(gdp->getPrimitiveRange();
    !it.atEnd(); ++it)
{
    GA_Offset offset = *it;
    if(gdp->getPrimitive(offset)->isDegenerate())
        bad_prims->addOffset(offset);
}
```

Attributes

Create Tuples

- `addFloatTuple()`
- `addIntTuple()`
- `addStringTuple()`
- `addDictTuple()`

Create Arrays

- `addFloatArray()`
- `addIntArray()`
- `addStringArray()`
- `addDictArray()`

Detached

- `createDetachedTupleAttribute()`

Attributes

GA_Detail

- findPointAttribute()
- findVertexAttribute()
- findPrimitiveAttribute()
- findGlobalAttribute()
- destroyAttribute()
- getAttributeDict()

GA_AttributeDict

- begin()
- end()

GA_Attribute

- getType()
- getName()
- getStorageClass()
- getTupleSize()

Attribute Type Info

```
GA_TYPE_VOID
GA_TYPE_POINT
GA_TYPE_HPOINT
GA_TYPE_VECTOR
GA_TYPE_NORMAL
GA_TYPE_COLOR
GA_TYPE_TRANSFORM
GA_TYPE_QUATERNION
GA_TYPE_INDEXPAIR
GA_TYPE_NONARITHMETIC_INTEGER
GA_TYPE_TEXTURE_COORD
```

GA_Attribute

- `getTypeInfo()`
- `setTypeInfo()`

Attribute Handle

F/D/I/M3/M4/Q/V3/S/Dict/FA/DA/IA/SA/DictA

GA_ROHandleF

```
- bind()
- isValid()
- get()
```

GA_RWHandleF

```
- set()
```

Example Reading Attribute Values

```
GA_ROHandleV3 uv_h(gdp, GA_ATTRIB_VERTEX, "uv");
UT_BoundingBox box;
box.initBounds();
if(uv_h.isValid())
{
    for(GA_Iterator it(gdp->getPrimitiveRange(group)); !it.atEnd(); ++it)
    {
        GA_Offset offset = *it;
        GA_Size nvtx = gdp->getPrimitiveVertexCount(offset);
        for(exint i = 0; i < nvtx; ++i)
            box.enlargeBounds(uv_h.get(gdp->getPrimitiveVertexOffset(offset, i)));
    }
}
```

Example Writing Attribute Values

```
GA_RWHandleS name_h(gdp, GA_ATTRIB_PRIMITIVE, "name");
if(!name_h.isValid())
    name_h.bind(gdp->addStringTuple(GA_ATTRIB_PRIMITIVE, "name", 1));
if(name_h.isValid())
{
    for(auto &iter : pieces)
    {
        const char *piece = iter.first.c_str();
        const GA_OffsetArray &offsets = iter.second;
        for(exint i = 0; i < offsets.entries(); ++i)
            name_h.set(offsets(i), piece);
    }
}
```

Example Writing Array Values

```
GA_RWHandleFA weights_h(gdp, GA_ATTRIB_POINT, "weights", 1);
if(!weights_h.isValid())
    weights_h.bind(gdp->addFloatArray(GA_ATTRIB_POINT, "weights", 1));
if(weights_h.isValid())
{
    UT_FloatArray w;
    for(GA_Iterator it(gdp->getPointRange(ptgroup)); !it.atEnd(); ++it)
    {
        GA_Offset pt = *it;
        weights_h.get(pt, w);
        for(exint i = 0; i < w.entries(); ++i)
            w(i) *= scale;
        weights_h.set(pt, w);
    }
}
```

Example Writing Dict Values

```
GA_RWHandleDict props_h(gdp, GA_ATTRIB_POINT, "properties", 1);
if(!props_h.isValid())
    props_h.bind(gdp->addDictTuple(GA_ATTRIB_POINT, "properties", 1));
if(props_h.isValid())
{
    UT_Options options;
    options.setOptionI("loops", 42);
    options.setOptionF("angle", 1.23);
    props_h.set(pt_offset, UT_OptionsHolder(&options));
}
```

String Table

GA_Attribute

- getAIFCopyData()
- getAIFCompare()
- getAIFMath()
- getAIFSharedStringTuple()

GA_AIFSharedStringTuple

- extractStrings()
- getTableString()
- getTableHandle()
- getHandle()
- getString()
- setHandle()
- setString()
- addStrings()

Attribute Wrangler

GA_ElementWrangler

- copyAttributeValues()
- addAttributeValues()
- scaleAttributeValues()
- lerpAttributeValues()

GA_AttributeFilter

- selectPublic()
- selectByPattern()
- selectAnd()
- selectOr()
- selectNot()

GA_PointWrangler
GA_VertexWrangler
GA_PrimitiveWrangler
GA_DetailWrangler

GA_Details

- cloneMissingAttributes()

Threading

Copy-on-Write

GA_PAGE_SIZE

- components/attributes/unorder groups

GA_SplittableRange

- `UTparallelFor()/UTparallelReduce()`

GA_PageHandle

- marshalled or raw access to underlying attribute data
- `setPage()`
- `get()/set()`
- `value()`

Example Multi-Threaded Update

```
GA_RWHandleV3 p_h(gdp->getP());
UTparallelFor(GA_SplittableRange(gdp->getPointRange()),
             [&](const GA_SplittableRange &r)
    {
        GA_Offset start, end;
        for (GA_Iterator it(r); it.blockAdvance(start, end); )
        {
            for (GA_Offset pt = start; pt < end; ++pt)
                p_h.set(pt, plane.project(p_h.get(pt)));
        }
    });
```

Example Multi-Threaded Update

```
UTparallelFor(GA_SplittableRange(gdp->getPointRange()),
             [&](const GA_SplittableRange &r)
    {
        GA_RWPageHandleV3 p_ph(gdp->getP());
        GA_Offset start, end;
        for (GA_Iterator it(r); it.blockAdvance(start, end); )
        {
            p_ph.setPage(start);
            plane.projectInPlace(&p_ph.value(start), end - start);
        }
    });
```

Packed Primitives

GU_PrimPacked

- `isPackedPrimitive()`
- `unpack()`
- `getPivot()/getPivot()`
- `viewportLOD()/setViewportLOD()`

```
GU_PackedGeometry::packGeometry()
GU_PackedDisk::packedDisk()
```

VDB

```
enum UT_VDBType
UTvdbGetGridType()
UTvdbConvert()
```

GU_PrimVDB

```
- build()
- buildFromPrimVolume()
- getStorageType()
- getConstGridPtr()
- setGrid()
```

```
- UTvdbCallRealType()
- UTvdbCallScalarType()
- UTvdbCallVec3Type()
- UTvdbCallPointType()
- UTvdbCallBoolType()
- UTvdbCallAllType()
- UTvdbCallAllTopology()
```

SOP

SOP_Node

- `myConstructor()`
- `buildTemplates()`
- `cookMySop()`
- `cookMyselfAsVerb()`
- `cookVerb()`
- `getDefaultState()`
- `inputLabel()`
- `isRefInput()`

SOP_NodeVerb

- `allocParms()`
- `allocCache()`
- `name()`
- `cookMode()`
- `cook()`
- `select()`

Caching

SOP_NodeVerb

- allocCache() - return a custom subclass of SOP_NodeCache

SOP_NodeCache

- getMemoryUsage()

SOP_NodeVerb::CookParms

- cache() - cast to custom subclass of SOP_NodeCache
- inputGeo()
- sopAddWarning()/sopAddError()

Caching

GA_Detail

- `getUniqueId()`
- `getMetaCacheCount()`

GA_Attribute

- `getDataId()`
- `bumpDataId()`
- `cloneDataId()`

SOP_Node

- `mySopFlags.setManagesDataIDs(true)`
- `bumpAllDataIds()`

GA_PrimitiveList / GA_Handle

- `bumpDataId()`
- `getDataId()`

GA_Topology

- `getDataId()`
- `cloneDataId()`

Performance Monitor

```
#include <UT/UT_PerfMonAutoEvent.h>

OP_Node *node = cookparms.getNode();
int nodeid = node ? node->getUniqueId() : -1;
UT_PerfMonAutoCookEvent event(nodeid, “Smoothing”);
```

Support Files

```
$ cd /opt/hfsX.Y.ZZZ
$ source houdini_setup
$ cd ~/HdkExamples
$ hython $HH/python3.7libs/generate_proto.py SOP_PolyClip.C SOP_PolyClip.proto.h
$ hcustom SOP_PolyClip.C
```

Operator Icon

- `HOUDINI_UI_ICON_PATH`
- eg. `$HOME/houdiniX.Y/config/Icons`

Help Card and Help Card Icon

- `HOUDINI_PATH/help`
- eg. `$HOME/houdiniX.Y/help`

HOUDINI_DSO_PATH

- ./dso
- $HOME/houdiniX.Y/dso
- $HOME/Library/Preferences/houdini/X.Y/dso (only on Mac OSX)
- /Users/Shared/houdini/X.Y/dso (only on Mac OSX)
- $HSITE/houdiniX.Y/dso
- $HFS/houdini/dso

- #include <UT/UT_DSOVersion.h>
- newSopOperator()

Debugging

- compile with debug information

```
hcustom -g SOP_PolyClip.C
```

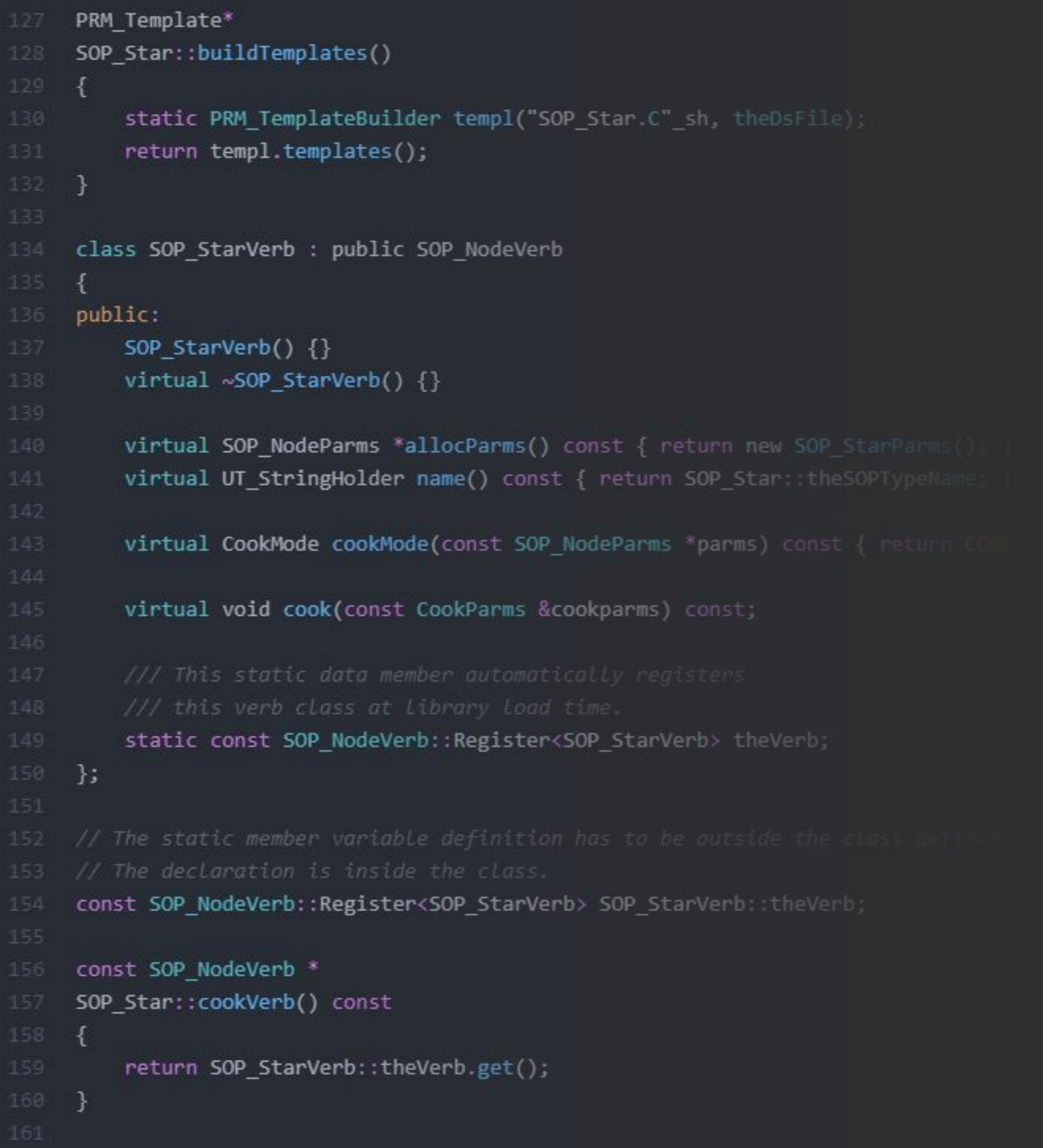
- running under a debugger

```
gdb $HFS/bin/houdini-bin
 (gdb) run -foreground
```

- assertions UT/UT_Assert.h

```
UT_ASSERT(expression)
```

- HOUDINI_DSO_ERROR



```cpp
PRM_Template*
SOP_Star::buildTemplates()
{
    static PRM_TemplateBuilder templ("SOP_Star.C"_sh, theDsFile);
    return templ.templates();
}

class SOP_StarVerb : public SOP_NodeVerb
{
public:
    SOP_StarVerb() {}
    virtual ~SOP_StarVerb() {}

    virtual SOP_NodeParms *allocParms() const { return new SOP_StarParms(); }
    virtual UT_StringHolder name() const { return SOP_Star::theSOPTypeName; }

    virtual CookMode cookMode(const SOP_NodeParms *parms) const { return COOK

    virtual void cook(const CookParms &cookparms) const;

    /// This static data member automatically registers
    /// this verb class at library load time.
    static const SOP_NodeVerb::Register<SOP_StarVerb> theVerb;
};

// The static member variable definition has to be outside the class defini
// The declaration is inside the class.
const SOP_NodeVerb::Register<SOP_StarVerb> SOP_StarVerb::theVerb;

const SOP_NodeVerb *
SOP_Star::cookVerb() const
{
    return SOP_StarVerb::theVerb.get();
}
```

THANK YOU

Web: SideFX.com

Twitter: sidefx

Facebook: Houdini3D

SideFX®