

## Tellusim Core SDK

Generated by Doxygen 1.8.13

## Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>2</b>
2.1	Class Hierarchy . . . . .	2
<b>3</b>	<b>Class Index</b>	<b>11</b>
3.1	Class List . . . . .	11
<b>4</b>	<b>Namespace Documentation</b>	<b>21</b>
4.1	Tellusim::Allocator Namespace Reference . . . . .	21
4.1.1	Detailed Description . . . . .	21
4.1.2	Function Documentation . . . . .	21
4.2	Tellusim::Android Namespace Reference . . . . .	22
4.2.1	Detailed Description . . . . .	22
4.3	Tellusim::Basis Namespace Reference . . . . .	22
4.3.1	Detailed Description . . . . .	22
4.4	Tellusim::Emscripten Namespace Reference . . . . .	23
4.4.1	Detailed Description . . . . .	23
4.5	Tellusim::Expression Namespace Reference . . . . .	23
4.5.1	Detailed Description . . . . .	24
4.6	Tellusim::iOS Namespace Reference . . . . .	24
4.6.1	Detailed Description . . . . .	24
4.7	Tellusim::Log Namespace Reference . . . . .	25
4.7.1	Detailed Description . . . . .	25
4.8	Tellusim::LU Namespace Reference . . . . .	25
4.8.1	Detailed Description . . . . .	25
4.9	Tellusim::MeshGraph Namespace Reference . . . . .	26
4.9.1	Detailed Description . . . . .	26
4.9.2	Function Documentation . . . . .	26

4.10 Tellusim::MeshReduce Namespace Reference . . . . .	26
4.10.1 Detailed Description . . . . .	27
4.10.2 Function Documentation . . . . .	27
4.11 Tellusim::MeshRefine Namespace Reference . . . . .	27
4.11.1 Detailed Description . . . . .	27
4.11.2 Function Documentation . . . . .	27
4.12 Tellusim::MeshSolid Namespace Reference . . . . .	28
4.12.1 Detailed Description . . . . .	28
4.12.2 Function Documentation . . . . .	28
4.13 Tellusim::Order Namespace Reference . . . . .	29
4.13.1 Detailed Description . . . . .	29
4.14 Tellusim::Parser Namespace Reference . . . . .	29
4.14.1 Detailed Description . . . . .	30
4.14.2 Function Documentation . . . . .	30
4.15 Tellusim::QR Namespace Reference . . . . .	33
4.15.1 Detailed Description . . . . .	33
4.16 Tellusim::Quadrilateral Namespace Reference . . . . .	33
4.16.1 Detailed Description . . . . .	33
4.17 Tellusim::Spatial Namespace Reference . . . . .	33
4.17.1 Detailed Description . . . . .	34
4.18 Tellusim::SVD Namespace Reference . . . . .	34
4.18.1 Detailed Description . . . . .	34
4.19 Tellusim::System Namespace Reference . . . . .	35
4.19.1 Detailed Description . . . . .	35
4.19.2 Function Documentation . . . . .	35
4.20 Tellusim::Time Namespace Reference . . . . .	36
4.20.1 Detailed Description . . . . .	36
4.21 Tellusim::Triangle Namespace Reference . . . . .	36
4.21.1 Detailed Description . . . . .	36
4.22 Tellusim::WinApp Namespace Reference . . . . .	36
4.22.1 Detailed Description . . . . .	37
4.23 Tellusim::Windows Namespace Reference . . . . .	37
4.23.1 Detailed Description . . . . .	38

<b>5</b>	<b>Class Documentation</b>	<b>38</b>
5.1	Tellusim::App Class Reference	38
5.1.1	Detailed Description	39
5.2	Tellusim::Archive Class Reference	39
5.2.1	Detailed Description	39
5.3	Tellusim::ArchiveStream Class Reference	40
5.3.1	Detailed Description	40
5.4	Tellusim::Async Class Reference	40
5.4.1	Detailed Description	42
5.5	Tellusim::AsyncFunction< Func > Class Template Reference	42
5.5.1	Detailed Description	43
5.6	Tellusim::AsyncFunctionBase Class Reference	43
5.6.1	Detailed Description	43
5.7	Tellusim::AsyncFunctionRet< Type > Struct Template Reference	43
5.7.1	Detailed Description	44
5.8	Tellusim::AsyncFunctionRet< void > Struct Template Reference	44
5.9	Tellusim::Atlas< Type > Class Template Reference	44
5.9.1	Detailed Description	45
5.10	Tellusim::AtomicArray< Type > Class Template Reference	45
5.10.1	Detailed Description	46
5.11	Tellusim::Atomici32 Struct Reference	46
5.11.1	Detailed Description	46
5.12	Tellusim::AtomicLock Struct Reference	47
5.12.1	Detailed Description	47
5.13	Tellusim::AtomicPtr< Type > Struct Template Reference	47
5.13.1	Detailed Description	48
5.14	Tellusim::BitonicSort Class Reference	48
5.14.1	Detailed Description	49
5.14.2	Member Function Documentation	49
5.15	Tellusim::Blob Class Reference	52

5.15.1 Detailed Description . . . . .	53
5.16 Tellusim::BoundingBox< T, Vector3 > Struct Template Reference . . . . .	53
5.16.1 Detailed Description . . . . .	56
5.17 Tellusim::BoundCircle< T, Vector2 > Struct Template Reference . . . . .	56
5.17.1 Detailed Description . . . . .	58
5.18 Tellusim::BoundFrustum< T > Struct Template Reference . . . . .	58
5.18.1 Detailed Description . . . . .	61
5.19 Tellusim::BoundRect< T, Vector2 > Struct Template Reference . . . . .	61
5.19.1 Detailed Description . . . . .	63
5.20 Tellusim::BoundSphere< T, Vector3 > Struct Template Reference . . . . .	63
5.20.1 Detailed Description . . . . .	66
5.21 Tellusim::BrepModel Class Reference . . . . .	66
5.21.1 Detailed Description . . . . .	68
5.22 Tellusim::Buffer Class Reference . . . . .	68
5.22.1 Detailed Description . . . . .	70
5.23 Tellusim::BufferTable Class Reference . . . . .	70
5.23.1 Detailed Description . . . . .	70
5.24 Tellusim::Tracing::BuildIndirect Struct Reference . . . . .	71
5.24.1 Detailed Description . . . . .	71
5.25 Tellusim::Canvas Class Reference . . . . .	71
5.25.1 Detailed Description . . . . .	73
5.26 Tellusim::CanvasElement Class Reference . . . . .	74
5.26.1 Detailed Description . . . . .	77
5.27 Tellusim::CanvasEllipse Class Reference . . . . .	77
5.27.1 Detailed Description . . . . .	79
5.28 Tellusim::CanvasMesh Class Reference . . . . .	79
5.28.1 Detailed Description . . . . .	81
5.29 Tellusim::CanvasRect Class Reference . . . . .	81
5.29.1 Detailed Description . . . . .	82
5.30 Tellusim::CanvasShape Class Reference . . . . .	82

5.30.1 Detailed Description . . . . .	84
5.31 Tellusim::CanvasShapeVertex Struct Reference . . . . .	84
5.31.1 Detailed Description . . . . .	84
5.32 Tellusim::CanvasStrip Class Reference . . . . .	85
5.32.1 Detailed Description . . . . .	86
5.33 Tellusim::CanvasStripVertex Struct Reference . . . . .	86
5.33.1 Detailed Description . . . . .	88
5.34 Tellusim::CanvasText Class Reference . . . . .	88
5.34.1 Detailed Description . . . . .	89
5.35 Tellusim::CanvasTriangle Class Reference . . . . .	89
5.35.1 Detailed Description . . . . .	90
5.36 Tellusim::CanvasVertex Struct Reference . . . . .	91
5.36.1 Detailed Description . . . . .	92
5.37 Tellusim::Color Struct Reference . . . . .	92
5.37.1 Detailed Description . . . . .	94
5.38 Tellusim::Command Class Reference . . . . .	94
5.38.1 Detailed Description . . . . .	97
5.39 Tellusim::Compute Class Reference . . . . .	98
5.39.1 Detailed Description . . . . .	100
5.40 Tellusim::RadixMap< Key, Type, Size >::ConstIterator Class Reference . . . . .	101
5.40.1 Detailed Description . . . . .	101
5.41 Tellusim::Context Class Reference . . . . .	101
5.41.1 Detailed Description . . . . .	102
5.42 Tellusim::Control Class Reference . . . . .	103
5.42.1 Detailed Description . . . . .	107
5.43 Tellusim::ControlArea Class Reference . . . . .	107
5.43.1 Detailed Description . . . . .	109
5.44 Tellusim::ControlBase Class Reference . . . . .	109
5.44.1 Detailed Description . . . . .	111
5.45 Tellusim::ControlButton Class Reference . . . . .	111

5.45.1 Detailed Description . . . . .	113
5.46 Tellusim::ControlCheck Class Reference . . . . .	113
5.46.1 Detailed Description . . . . .	114
5.47 Tellusim::ControlCombo Class Reference . . . . .	114
5.47.1 Detailed Description . . . . .	117
5.48 Tellusim::ControlDialog Class Reference . . . . .	117
5.48.1 Detailed Description . . . . .	119
5.49 Tellusim::ControlEdit Class Reference . . . . .	119
5.49.1 Detailed Description . . . . .	121
5.50 Tellusim::ControlGrid Class Reference . . . . .	121
5.50.1 Detailed Description . . . . .	122
5.51 Tellusim::ControlGroup Class Reference . . . . .	122
5.51.1 Detailed Description . . . . .	124
5.52 Tellusim::ControlPanel Class Reference . . . . .	125
5.52.1 Detailed Description . . . . .	126
5.53 Tellusim::ControlRect Class Reference . . . . .	126
5.53.1 Detailed Description . . . . .	129
5.54 Tellusim::ControlRoot Class Reference . . . . .	129
5.54.1 Detailed Description . . . . .	132
5.55 Tellusim::ControlScroll Class Reference . . . . .	132
5.55.1 Detailed Description . . . . .	134
5.56 Tellusim::ControlSlider Class Reference . . . . .	134
5.56.1 Detailed Description . . . . .	136
5.56.2 Member Function Documentation . . . . .	136
5.57 Tellusim::ControlSplit Class Reference . . . . .	137
5.57.1 Detailed Description . . . . .	138
5.58 Tellusim::ControlText Class Reference . . . . .	139
5.58.1 Detailed Description . . . . .	140
5.59 Tellusim::ControlTree Class Reference . . . . .	141
5.59.1 Detailed Description . . . . .	144

5.60	Tellusim::ControlWindow Class Reference	144
5.60.1	Detailed Description	145
5.61	Tellusim::CubeFilter Class Reference	145
5.61.1	Detailed Description	146
5.61.2	Member Function Documentation	146
5.62	Tellusim::CUBuffer Class Reference	147
5.62.1	Detailed Description	148
5.63	Tellusim::CUContext Class Reference	149
5.63.1	Detailed Description	150
5.64	Tellusim::CUShader Class Reference	150
5.64.1	Detailed Description	150
5.65	Tellusim::CUTexture Class Reference	151
5.65.1	Detailed Description	151
5.66	Tellusim::D3D11Buffer Class Reference	152
5.66.1	Detailed Description	152
5.67	Tellusim::D3D11Context Class Reference	153
5.67.1	Detailed Description	154
5.68	Tellusim::D3D11Device Class Reference	154
5.68.1	Detailed Description	154
5.69	Tellusim::D3D11Shader Class Reference	155
5.69.1	Detailed Description	155
5.70	Tellusim::D3D11Surface Class Reference	156
5.70.1	Detailed Description	157
5.71	Tellusim::D3D11Target Class Reference	157
5.71.1	Detailed Description	158
5.72	Tellusim::D3D11Texture Class Reference	158
5.72.1	Detailed Description	159
5.73	Tellusim::D3D12Buffer Class Reference	159
5.73.1	Detailed Description	160
5.74	Tellusim::D3D12Command Class Reference	160



5.74.1 Detailed Description . . . . .	161
5.75 Tellusim::D3D12Compute Class Reference . . . . .	161
5.75.1 Detailed Description . . . . .	161
5.76 Tellusim::D3D12Context Class Reference . . . . .	162
5.76.1 Detailed Description . . . . .	163
5.77 Tellusim::D3D12Device Class Reference . . . . .	163
5.77.1 Detailed Description . . . . .	164
5.78 Tellusim::D3D12Tracing::D3D12Instance Struct Reference . . . . .	164
5.78.1 Detailed Description . . . . .	164
5.79 Tellusim::D3D12Kernel Class Reference . . . . .	164
5.79.1 Detailed Description . . . . .	165
5.80 Tellusim::D3D12Pipeline Class Reference . . . . .	165
5.80.1 Detailed Description . . . . .	166
5.81 Tellusim::D3D12Shader Class Reference . . . . .	166
5.81.1 Detailed Description . . . . .	167
5.82 Tellusim::D3D12Surface Class Reference . . . . .	167
5.82.1 Detailed Description . . . . .	168
5.83 Tellusim::D3D12Target Class Reference . . . . .	169
5.83.1 Detailed Description . . . . .	169
5.84 Tellusim::D3D12Texture Class Reference . . . . .	170
5.84.1 Detailed Description . . . . .	171
5.85 Tellusim::D3D12Tracing Class Reference . . . . .	171
5.85.1 Detailed Description . . . . .	172
5.86 Tellusim::D3D12Traversal Class Reference . . . . .	172
5.86.1 Detailed Description . . . . .	173
5.87 Tellusim::Date Class Reference . . . . .	173
5.87.1 Detailed Description . . . . .	173
5.87.2 Member Function Documentation . . . . .	174
5.88 Tellusim::DecoderJPEG Class Reference . . . . .	174
5.88.1 Detailed Description . . . . .	175

5.88.2 Member Function Documentation . . . . .	175
5.89 Tellusim::DefaultDestructor< Type > Struct Template Reference . . . . .	177
5.89.1 Detailed Description . . . . .	177
5.90 Tellusim::Desktop Class Reference . . . . .	177
5.90.1 Detailed Description . . . . .	178
5.91 Tellusim::Device Class Reference . . . . .	178
5.91.1 Detailed Description . . . . .	183
5.92 Tellusim::DialogColor Class Reference . . . . .	184
5.92.1 Detailed Description . . . . .	184
5.93 Tellusim::DialogDirectory Class Reference . . . . .	185
5.93.1 Detailed Description . . . . .	185
5.94 Tellusim::DialogFileOpen Class Reference . . . . .	185
5.94.1 Detailed Description . . . . .	186
5.95 Tellusim::DialogFileSave Class Reference . . . . .	186
5.95.1 Detailed Description . . . . .	187
5.96 Tellusim::DialogMenu Class Reference . . . . .	187
5.96.1 Detailed Description . . . . .	189
5.97 Tellusim::DialogMessage Class Reference . . . . .	189
5.97.1 Detailed Description . . . . .	190
5.98 Tellusim::DialogProgress Class Reference . . . . .	190
5.98.1 Detailed Description . . . . .	191
5.99 Tellusim::Directory Class Reference . . . . .	191
5.99.1 Detailed Description . . . . .	193
5.100 Tellusim::Compute::DispatchIndirect Struct Reference . . . . .	193
5.100.1 Detailed Description . . . . .	193
5.101 Tellusim::BitonicSort::DispatchParameters Struct Reference . . . . .	193
5.102 Tellusim::PrefixScan::DispatchParameters Struct Reference . . . . .	193
5.103 Tellusim::RadixSort::DispatchParameters Struct Reference . . . . .	194
5.104 Tellusim::SpatialGrid::DispatchParameters Struct Reference . . . . .	194
5.105 Tellusim::SpatialTree::DispatchParameters Struct Reference . . . . .	194

5.106Tellusim::Command::DrawArraysIndirect Struct Reference . . . . .	194
5.106.1 Detailed Description . . . . .	194
5.107Tellusim::Command::DrawElementsIndirect Struct Reference . . . . .	195
5.107.1 Detailed Description . . . . .	195
5.108Tellusim::Command::DrawMeshIndirect Struct Reference . . . . .	195
5.108.1 Detailed Description . . . . .	195
5.109Tellusim::EncoderASTC Class Reference . . . . .	195
5.109.1 Detailed Description . . . . .	196
5.109.2 Member Function Documentation . . . . .	196
5.110Tellusim::EncoderBC15 Class Reference . . . . .	197
5.110.1 Detailed Description . . . . .	198
5.110.2 Member Function Documentation . . . . .	198
5.111Tellusim::EncoderBC67 Class Reference . . . . .	198
5.111.1 Detailed Description . . . . .	199
5.111.2 Member Function Documentation . . . . .	199
5.112Tellusim::f32u32 Union Reference . . . . .	199
5.112.1 Detailed Description . . . . .	200
5.113Tellusim::f64u64 Union Reference . . . . .	200
5.113.1 Detailed Description . . . . .	200
5.114Tellusim::Face Struct Reference . . . . .	200
5.114.1 Detailed Description . . . . .	201
5.115Tellusim::BrepModel::FaceParameters Struct Reference . . . . .	201
5.116Tellusim::Device::Features Struct Reference . . . . .	202
5.116.1 Detailed Description . . . . .	204
5.117Tellusim::Fence Class Reference . . . . .	204
5.117.1 Detailed Description . . . . .	205
5.118Tellusim::File Class Reference . . . . .	205
5.118.1 Detailed Description . . . . .	206
5.119Tellusim::FixedPool< Type, Index > Class Template Reference . . . . .	206
5.119.1 Detailed Description . . . . .	207

5.120Tellusim::float16_t Struct Reference . . . . .	207
5.120.1 Detailed Description . . . . .	208
5.121Tellusim::float16x4_t Struct Reference . . . . .	209
5.121.1 Detailed Description . . . . .	210
5.121.2 Constructor & Destructor Documentation . . . . .	210
5.122Tellusim::float16x8_t Struct Reference . . . . .	210
5.122.1 Detailed Description . . . . .	212
5.122.2 Constructor & Destructor Documentation . . . . .	212
5.123Tellusim::float21_t Struct Reference . . . . .	212
5.123.1 Detailed Description . . . . .	213
5.124Tellusim::float24_t Struct Reference . . . . .	213
5.124.1 Detailed Description . . . . .	214
5.125Tellusim::float32x4_t Struct Reference . . . . .	214
5.125.1 Detailed Description . . . . .	215
5.125.2 Constructor & Destructor Documentation . . . . .	215
5.126Tellusim::float32x8_t Struct Reference . . . . .	216
5.126.1 Detailed Description . . . . .	217
5.126.2 Constructor & Destructor Documentation . . . . .	217
5.127Tellusim::float64x2_t Struct Reference . . . . .	217
5.127.1 Detailed Description . . . . .	218
5.128Tellusim::float64x4_t Struct Reference . . . . .	218
5.128.1 Detailed Description . . . . .	220
5.129Tellusim::float64x8_t Struct Reference . . . . .	220
5.129.1 Detailed Description . . . . .	221
5.130Tellusim::Font Class Reference . . . . .	221
5.130.1 Detailed Description . . . . .	222
5.131Tellusim::FontBatch Struct Reference . . . . .	223
5.132Tellusim::FontBatch32 Struct Reference . . . . .	224
5.133Tellusim::FontStyle Struct Reference . . . . .	225
5.133.1 Detailed Description . . . . .	226

5.134Tellusim::FourierTransform Class Reference . . . . .	226
5.134.1 Detailed Description . . . . .	227
5.134.2 Member Function Documentation . . . . .	227
5.135Tellusim::FUBuffer Class Reference . . . . .	228
5.135.1 Detailed Description . . . . .	228
5.136Tellusim::FUCommand Class Reference . . . . .	229
5.136.1 Detailed Description . . . . .	229
5.137Tellusim::FUCompute Class Reference . . . . .	230
5.137.1 Detailed Description . . . . .	230
5.138Tellusim::FUDevice Class Reference . . . . .	231
5.138.1 Detailed Description . . . . .	231
5.139Tellusim::FUFence Class Reference . . . . .	232
5.139.1 Detailed Description . . . . .	232
5.140Tellusim::FUKernel Class Reference . . . . .	233
5.140.1 Detailed Description . . . . .	233
5.141Tellusim::FUPipeline Class Reference . . . . .	234
5.141.1 Detailed Description . . . . .	234
5.142Tellusim::FUQuery Class Reference . . . . .	235
5.142.1 Detailed Description . . . . .	235
5.143Tellusim::FUSampler Class Reference . . . . .	236
5.143.1 Detailed Description . . . . .	236
5.144Tellusim::FUShader Class Reference . . . . .	237
5.144.1 Detailed Description . . . . .	237
5.145Tellusim::FUTarget Class Reference . . . . .	238
5.145.1 Detailed Description . . . . .	238
5.146Tellusim::FUTexture Class Reference . . . . .	239
5.146.1 Detailed Description . . . . .	239
5.147Tellusim::FUTracing Class Reference . . . . .	240
5.147.1 Detailed Description . . . . .	240
5.148Tellusim::FUTraversal Class Reference . . . . .	241

5.148.1 Detailed Description . . . . .	241
5.149Tellusim::GLBuffer Class Reference . . . . .	242
5.149.1 Detailed Description . . . . .	242
5.150Tellusim::GLContext Class Reference . . . . .	243
5.150.1 Detailed Description . . . . .	244
5.151Tellusim::GLSBuffer Class Reference . . . . .	244
5.151.1 Detailed Description . . . . .	244
5.152Tellusim::GLSContext Class Reference . . . . .	245
5.152.1 Detailed Description . . . . .	246
5.153Tellusim::GLSShader Class Reference . . . . .	246
5.153.1 Detailed Description . . . . .	246
5.154Tellusim::GLSSurface Class Reference . . . . .	247
5.154.1 Detailed Description . . . . .	248
5.155Tellusim::GLSTarget Class Reference . . . . .	248
5.155.1 Detailed Description . . . . .	248
5.156Tellusim::GLSTexture Class Reference . . . . .	249
5.156.1 Detailed Description . . . . .	249
5.157Tellusim::GLShader Class Reference . . . . .	250
5.157.1 Detailed Description . . . . .	250
5.158Tellusim::GLSurface Class Reference . . . . .	251
5.158.1 Detailed Description . . . . .	252
5.159Tellusim::GLTarget Class Reference . . . . .	252
5.159.1 Detailed Description . . . . .	252
5.160Tellusim::GLTexture Class Reference . . . . .	253
5.160.1 Detailed Description . . . . .	253
5.161Tellusim::GradientStyle Struct Reference . . . . .	254
5.161.1 Detailed Description . . . . .	255
5.162Tellusim::HeapPool< Threshold > Class Template Reference . . . . .	255
5.162.1 Detailed Description . . . . .	255
5.163Tellusim::HIPBuffer Class Reference . . . . .	256

5.163.1 Detailed Description . . . . .	256
5.164Tellusim::HIPContext Class Reference . . . . .	257
5.164.1 Detailed Description . . . . .	258
5.165Tellusim::HIPShader Class Reference . . . . .	258
5.165.1 Detailed Description . . . . .	258
5.166Tellusim::HIPTexture Class Reference . . . . .	259
5.166.1 Detailed Description . . . . .	259
5.167Tellusim::Image Class Reference . . . . .	259
5.167.1 Detailed Description . . . . .	263
5.168Tellusim::ImageColor Struct Reference . . . . .	264
5.168.1 Detailed Description . . . . .	265
5.169Tellusim::ImageSampler Class Reference . . . . .	265
5.169.1 Detailed Description . . . . .	267
5.170Tellusim::ImageStream Class Reference . . . . .	267
5.170.1 Detailed Description . . . . .	268
5.171Tellusim::Info Class Reference . . . . .	268
5.171.1 Detailed Description . . . . .	268
5.172Tellusim::Tracing::Instance Struct Reference . . . . .	269
5.172.1 Detailed Description . . . . .	269
5.173Tellusim::int16x8_t Struct Reference . . . . .	269
5.173.1 Detailed Description . . . . .	271
5.173.2 Constructor & Destructor Documentation . . . . .	271
5.174Tellusim::int32x4_t Struct Reference . . . . .	271
5.174.1 Detailed Description . . . . .	273
5.174.2 Constructor & Destructor Documentation . . . . .	273
5.175Tellusim::int32x8_t Struct Reference . . . . .	273
5.175.1 Detailed Description . . . . .	275
5.175.2 Constructor & Destructor Documentation . . . . .	275
5.176Tellusim::IsPtr< Type > Struct Template Reference . . . . .	275
5.176.1 Detailed Description . . . . .	275

5.177Tellusim::RadixMap< Key, Type, Size >::Iterator Class Reference . . . . .	276
5.177.1 Detailed Description . . . . .	276
5.178Tellusim::Json Class Reference . . . . .	276
5.178.1 Detailed Description . . . . .	279
5.179Tellusim::Kernel Class Reference . . . . .	279
5.179.1 Detailed Description . . . . .	281
5.180Tellusim::MeshTransform::KeyData< Type > Struct Template Reference . . . . .	281
5.180.1 Detailed Description . . . . .	281
5.181Tellusim::Layer Struct Reference . . . . .	282
5.182Tellusim::Matrix3x2< T > Struct Template Reference . . . . .	282
5.182.1 Detailed Description . . . . .	285
5.183Tellusim::Matrix4x3< T > Struct Template Reference . . . . .	285
5.183.1 Detailed Description . . . . .	289
5.184Tellusim::Matrix4x4< T > Struct Template Reference . . . . .	289
5.184.1 Detailed Description . . . . .	293
5.185Tellusim::MatrixNxM< Type, N, M > Struct Template Reference . . . . .	293
5.185.1 Detailed Description . . . . .	295
5.186Tellusim::Mesh Class Reference . . . . .	295
5.186.1 Detailed Description . . . . .	298
5.186.2 Member Function Documentation . . . . .	298
5.187Tellusim::MeshAnimation Class Reference . . . . .	300
5.187.1 Detailed Description . . . . .	302
5.188Tellusim::MeshAttachment Class Reference . . . . .	302
5.188.1 Detailed Description . . . . .	303
5.189Tellusim::MeshAttribute Class Reference . . . . .	304
5.189.1 Detailed Description . . . . .	306
5.190Tellusim::MeshGeometry Class Reference . . . . .	306
5.190.1 Detailed Description . . . . .	309
5.190.2 Member Function Documentation . . . . .	309
5.191Tellusim::MeshIndices Class Reference . . . . .	312



5.191.1 Detailed Description . . . . .	314
5.192Tellusim::MeshJoint Class Reference . . . . .	314
5.192.1 Detailed Description . . . . .	315
5.193Tellusim::MeshModel::Meshlet Struct Reference . . . . .	315
5.194Tellusim::MeshMaterial Class Reference . . . . .	315
5.194.1 Detailed Description . . . . .	317
5.195Tellusim::MeshModel Class Reference . . . . .	317
5.195.1 Detailed Description . . . . .	319
5.195.2 Member Function Documentation . . . . .	320
5.196Tellusim::MeshNode Class Reference . . . . .	321
5.196.1 Detailed Description . . . . .	322
5.197Tellusim::MeshStream Class Reference . . . . .	322
5.197.1 Detailed Description . . . . .	323
5.198Tellusim::MeshTransform Class Reference . . . . .	323
5.198.1 Detailed Description . . . . .	324
5.199Tellusim::Mipmap Struct Reference . . . . .	324
5.200Tellusim::MTLBuffer Class Reference . . . . .	325
5.200.1 Detailed Description . . . . .	325
5.201Tellusim::MTLCommand Class Reference . . . . .	326
5.201.1 Detailed Description . . . . .	326
5.202Tellusim::MTLCompute Class Reference . . . . .	327
5.202.1 Detailed Description . . . . .	327
5.203Tellusim::MTLContext Class Reference . . . . .	328
5.203.1 Detailed Description . . . . .	329
5.204Tellusim::MTLDevice Class Reference . . . . .	329
5.204.1 Detailed Description . . . . .	330
5.205Tellusim::MTLTracing::MTLInstance Struct Reference . . . . .	330
5.205.1 Detailed Description . . . . .	330
5.206Tellusim::MTLKernel Class Reference . . . . .	330
5.206.1 Detailed Description . . . . .	331

5.207Tellusim::MTLPipeline Class Reference . . . . .	331
5.207.1 Detailed Description . . . . .	332
5.208Tellusim::MTLSampler Class Reference . . . . .	332
5.208.1 Detailed Description . . . . .	333
5.209Tellusim::MTLShader Class Reference . . . . .	333
5.209.1 Detailed Description . . . . .	334
5.210Tellusim::MTLSurface Class Reference . . . . .	334
5.210.1 Detailed Description . . . . .	335
5.211Tellusim::MTLTarget Class Reference . . . . .	335
5.211.1 Detailed Description . . . . .	336
5.212Tellusim::MTLTexture Class Reference . . . . .	336
5.212.1 Detailed Description . . . . .	337
5.213Tellusim::MTLTracing Class Reference . . . . .	337
5.213.1 Detailed Description . . . . .	338
5.214Tellusim::Spatial::Node< Type, Size > Struct Template Reference . . . . .	338
5.214.1 Detailed Description . . . . .	339
5.215Tellusim::Atlas< Type >::Node Struct Reference . . . . .	339
5.215.1 Detailed Description . . . . .	339
5.216Tellusim::SpatialTree::Node Struct Reference . . . . .	340
5.217Tellusim::Origin Struct Reference . . . . .	340
5.217.1 Detailed Description . . . . .	340
5.218Tellusim::Pipeline Class Reference . . . . .	340
5.218.1 Detailed Description . . . . .	347
5.219Tellusim::Polygon< Capacity > Struct Template Reference . . . . .	347
5.219.1 Detailed Description . . . . .	348
5.219.2 Member Function Documentation . . . . .	348
5.220Tellusim::PrefixScan Class Reference . . . . .	349
5.220.1 Detailed Description . . . . .	350
5.220.2 Member Function Documentation . . . . .	350
5.221Tellusim::Quaternion< T > Struct Template Reference . . . . .	353

5.221.1 Detailed Description . . . . .	356
5.222Tellusim::Query Class Reference . . . . .	356
5.222.1 Detailed Description . . . . .	357
5.223Tellusim::RadixCompare< Type > Struct Template Reference . . . . .	357
5.223.1 Detailed Description . . . . .	357
5.224Tellusim::RadixCompare< float32_t > Struct Template Reference . . . . .	358
5.225Tellusim::RadixCompare< int32_t > Struct Template Reference . . . . .	358
5.226Tellusim::RadixCompare< uint32_t > Struct Template Reference . . . . .	358
5.227Tellusim::RadixMap< Key, Type, Size > Class Template Reference . . . . .	358
5.227.1 Detailed Description . . . . .	359
5.228Tellusim::RadixSort Class Reference . . . . .	359
5.228.1 Detailed Description . . . . .	361
5.228.2 Member Function Documentation . . . . .	361
5.229Tellusim::Random< Integer, Float > Struct Template Reference . . . . .	363
5.229.1 Detailed Description . . . . .	364
5.230Tellusim::Rect Struct Reference . . . . .	364
5.230.1 Detailed Description . . . . .	365
5.231Tellusim::Region Struct Reference . . . . .	366
5.231.1 Detailed Description . . . . .	366
5.232Tellusim::Sampler Class Reference . . . . .	366
5.232.1 Detailed Description . . . . .	368
5.233Tellusim::Scissor Struct Reference . . . . .	368
5.233.1 Detailed Description . . . . .	369
5.234Tellusim::SeparableFilter Class Reference . . . . .	369
5.234.1 Detailed Description . . . . .	370
5.234.2 Member Function Documentation . . . . .	370
5.235Tellusim::Shader Class Reference . . . . .	371
5.235.1 Detailed Description . . . . .	375
5.236Tellusim::ShaderCompiler Class Reference . . . . .	375
5.237Tellusim::Size Struct Reference . . . . .	376

5.237.1 Detailed Description . . . . .	376
5.238Tellusim::Slice Struct Reference . . . . .	377
5.239Tellusim::Socket Class Reference . . . . .	377
5.239.1 Detailed Description . . . . .	379
5.240Tellusim::SocketSSL Class Reference . . . . .	379
5.240.1 Detailed Description . . . . .	380
5.241Tellusim::Source Class Reference . . . . .	380
5.241.1 Detailed Description . . . . .	382
5.242Tellusim::SpatialGrid Class Reference . . . . .	382
5.242.1 Detailed Description . . . . .	382
5.242.2 Member Function Documentation . . . . .	382
5.243Tellusim::SpatialTree Class Reference . . . . .	383
5.243.1 Detailed Description . . . . .	385
5.243.2 Member Function Documentation . . . . .	385
5.244Tellusim::SpinLock Struct Reference . . . . .	387
5.244.1 Detailed Description . . . . .	388
5.245Tellusim::Query::Statistics Struct Reference . . . . .	388
5.245.1 Detailed Description . . . . .	388
5.246Tellusim::Stream Class Reference . . . . .	388
5.246.1 Detailed Description . . . . .	390
5.247Tellusim::String Class Reference . . . . .	391
5.247.1 Detailed Description . . . . .	394
5.248Tellusim::StrokeStyle Struct Reference . . . . .	394
5.248.1 Detailed Description . . . . .	395
5.249Tellusim::Surface Class Reference . . . . .	396
5.249.1 Detailed Description . . . . .	397
5.250Tellusim::Target Class Reference . . . . .	397
5.250.1 Detailed Description . . . . .	399
5.251Tellusim::Async::Task Class Reference . . . . .	399
5.251.1 Detailed Description . . . . .	400

5.252Tellusim::Tensor Struct Reference . . . . .	400
5.252.1 Detailed Description . . . . .	402
5.253Tellusim::TensorGraph Class Reference . . . . .	402
5.253.1 Detailed Description . . . . .	404
5.253.2 Member Function Documentation . . . . .	404
5.254Tellusim::Texture Class Reference . . . . .	405
5.254.1 Detailed Description . . . . .	408
5.255Tellusim::TextureTable Class Reference . . . . .	408
5.255.1 Detailed Description . . . . .	409
5.256Tellusim::Thread Class Reference . . . . .	409
5.256.1 Detailed Description . . . . .	410
5.257Tellusim::ThreadClassFunction0< Class, Func > Class Template Reference . . . . .	410
5.257.1 Detailed Description . . . . .	411
5.258Tellusim::ThreadClassFunction1< Class, Func, A0 > Class Template Reference . . . . .	411
5.259Tellusim::ThreadClassFunction2< Class, Func, A0, A1 > Class Template Reference . . . . .	412
5.260Tellusim::ThreadClassFunction3< Class, Func, A0, A1, A2 > Class Template Reference . . . . .	413
5.261Tellusim::ThreadClassFunction4< Class, Func, A0, A1, A2, A3 > Class Template Reference . . . . .	414
5.262Tellusim::ThreadFunction0< Func > Class Template Reference . . . . .	415
5.262.1 Detailed Description . . . . .	416
5.263Tellusim::ThreadFunction1< Func, A0 > Class Template Reference . . . . .	416
5.264Tellusim::ThreadFunction2< Func, A0, A1 > Class Template Reference . . . . .	417
5.265Tellusim::ThreadFunction3< Func, A0, A1, A2 > Class Template Reference . . . . .	418
5.266Tellusim::ThreadFunction4< Func, A0, A1, A2, A3 > Class Template Reference . . . . .	419
5.267Tellusim::Tracing Class Reference . . . . .	420
5.267.1 Detailed Description . . . . .	423
5.268Tellusim::Traversal Class Reference . . . . .	423
5.268.1 Detailed Description . . . . .	425
5.269Tellusim::uint16x8_t Struct Reference . . . . .	426
5.269.1 Detailed Description . . . . .	427
5.269.2 Constructor & Destructor Documentation . . . . .	427

5.270Tellusim::uint32x4_t Struct Reference . . . . .	427
5.270.1 Detailed Description . . . . .	429
5.270.2 Constructor & Destructor Documentation . . . . .	429
5.271Tellusim::uint32x8_t Struct Reference . . . . .	429
5.271.1 Detailed Description . . . . .	431
5.271.2 Constructor & Destructor Documentation . . . . .	431
5.272Tellusim::Vector2< T > Struct Template Reference . . . . .	431
5.272.1 Detailed Description . . . . .	433
5.273Tellusim::Vector3< T > Struct Template Reference . . . . .	433
5.273.1 Detailed Description . . . . .	436
5.274Tellusim::Vector4< T > Struct Template Reference . . . . .	436
5.274.1 Detailed Description . . . . .	439
5.275Tellusim::VectorN< Type, N > Struct Template Reference . . . . .	439
5.275.1 Detailed Description . . . . .	440
5.276Tellusim::Viewport Struct Reference . . . . .	440
5.276.1 Detailed Description . . . . .	441
5.277Tellusim::VKBuffer Class Reference . . . . .	441
5.277.1 Detailed Description . . . . .	442
5.278Tellusim::VKCommand Class Reference . . . . .	442
5.278.1 Detailed Description . . . . .	443
5.279Tellusim::VKCompute Class Reference . . . . .	443
5.279.1 Detailed Description . . . . .	444
5.280Tellusim::VKContext Class Reference . . . . .	444
5.280.1 Detailed Description . . . . .	445
5.281Tellusim::VKDevice Class Reference . . . . .	445
5.281.1 Detailed Description . . . . .	446
5.282Tellusim::VKFence Class Reference . . . . .	447
5.282.1 Detailed Description . . . . .	447
5.283Tellusim::VKTracing::VKInstance Struct Reference . . . . .	447
5.283.1 Detailed Description . . . . .	448

5.284Tellusim::VKShader Class Reference . . . . .	448
5.284.1 Detailed Description . . . . .	449
5.285Tellusim::VKSurface Class Reference . . . . .	449
5.285.1 Detailed Description . . . . .	450
5.286Tellusim::VKTarget Class Reference . . . . .	450
5.286.1 Detailed Description . . . . .	451
5.287Tellusim::VKTexture Class Reference . . . . .	451
5.287.1 Detailed Description . . . . .	452
5.288Tellusim::VKTracing Class Reference . . . . .	452
5.288.1 Detailed Description . . . . .	453
5.289Tellusim::Window Class Reference . . . . .	453
5.289.1 Detailed Description . . . . .	459
5.290Tellusim::Xml Class Reference . . . . .	459
5.290.1 Detailed Description . . . . .	462
<b>Index</b>	<b>463</b>

## 1 Namespace Index

### 1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<b>Tellusim::Allocator</b>	<b>21</b>
<b>Tellusim::Android</b>	<b>22</b>
<b>Tellusim::Basis</b>	<b>22</b>
<b>Tellusim::Emscripten</b>	<b>23</b>
<b>Tellusim::Expression</b>	<b>23</b>
<b>Tellusim::iOS</b>	<b>24</b>
<b>Tellusim::Log</b>	<b>25</b>
<b>Tellusim::LU</b>	<b>25</b>
<b>Tellusim::MeshGraph</b>	<b>26</b>
<b>Tellusim::MeshReduce</b>	<b>26</b>

<a href="#">Tellusim::MeshRefine</a>	27
<a href="#">Tellusim::MeshSolid</a>	28
<a href="#">Tellusim::Order</a>	29
<a href="#">Tellusim::Parser</a>	29
<a href="#">Tellusim::QR</a>	33
<a href="#">Tellusim::Quadrilateral</a>	33
<a href="#">Tellusim::Spatial</a>	33
<a href="#">Tellusim::SVD</a>	34
<a href="#">Tellusim::System</a>	35
<a href="#">Tellusim::Time</a>	36
<a href="#">Tellusim::Triangle</a>	36
<a href="#">Tellusim::WinApp</a>	36
<a href="#">Tellusim::Windows</a>	37

## 2 Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

<a href="#">Tellusim::App</a>	38
<a href="#">Tellusim::Archive</a>	39
<a href="#">Tellusim::ArchiveStream</a>	40
<a href="#">Tellusim::Async</a>	40
<a href="#">Tellusim::AsyncFunctionBase</a>	43
<a href="#">Tellusim::AsyncFunction&lt; Func &gt;</a>	42
<a href="#">Tellusim::AsyncFunctionRet&lt; Type &gt;</a>	43
<a href="#">Tellusim::AsyncFunctionRet&lt; Ret &gt;</a>	43
<a href="#">Tellusim::AsyncFunctionRet&lt; void &gt;</a>	44
<a href="#">Tellusim::Atlas&lt; Type &gt;</a>	44
<a href="#">Tellusim::AtomicArray&lt; Type &gt;</a>	45
<a href="#">Tellusim::Atomici32</a>	46
<a href="#">Tellusim::AtomicLock</a>	47
<a href="#">Tellusim::AtomicPtr&lt; Type &gt;</a>	47



<b>Tellusim::BitonicSort</b>	<b>48</b>
<b>Tellusim::BoundingBox&lt; T, Vector3 &gt;</b>	<b>53</b>
<b>Tellusim::BoundCircle&lt; T, Vector2 &gt;</b>	<b>56</b>
<b>Tellusim::BoundFrustum&lt; T &gt;</b>	<b>58</b>
<b>Tellusim::BoundRect&lt; T, Vector2 &gt;</b>	<b>61</b>
<b>Tellusim::BoundSphere&lt; T, Vector3 &gt;</b>	<b>63</b>
<b>Tellusim::BrepModel</b>	<b>66</b>
<b>Tellusim::Buffer</b>	<b>68</b>
<b>Tellusim::CUBuffer</b>	<b>147</b>
<b>Tellusim::D3D11Buffer</b>	<b>152</b>
<b>Tellusim::D3D12Buffer</b>	<b>159</b>
<b>Tellusim::FUBuffer</b>	<b>228</b>
<b>Tellusim::GLBuffer</b>	<b>242</b>
<b>Tellusim::GLESBuffer</b>	<b>244</b>
<b>Tellusim::HIPBuffer</b>	<b>256</b>
<b>Tellusim::MTLBuffer</b>	<b>325</b>
<b>Tellusim::VKBuffer</b>	<b>441</b>
<b>Tellusim::BufferTable</b>	<b>70</b>
<b>Tellusim::Tracing::BuildIndirect</b>	<b>71</b>
<b>Tellusim::Canvas</b>	<b>71</b>
<b>Tellusim::CanvasElement</b>	<b>74</b>
<b>Tellusim::CanvasEllipse</b>	<b>77</b>
<b>Tellusim::CanvasMesh</b>	<b>79</b>
<b>Tellusim::CanvasRect</b>	<b>81</b>
<b>Tellusim::CanvasShape</b>	<b>82</b>
<b>Tellusim::CanvasStrip</b>	<b>85</b>
<b>Tellusim::CanvasText</b>	<b>88</b>
<b>Tellusim::CanvasTriangle</b>	<b>89</b>
<b>Tellusim::CanvasShapeVertex</b>	<b>84</b>
<b>Tellusim::CanvasStripVertex</b>	<b>86</b>
<b>Tellusim::CanvasVertex</b>	<b>91</b>
<b>Tellusim::Color</b>	<b>92</b>

<b>Tellusim::Command</b>	<b>94</b>
<b>Tellusim::D3D12Command</b>	<b>160</b>
<b>Tellusim::FUCommand</b>	<b>229</b>
<b>Tellusim::MTLCommand</b>	<b>326</b>
<b>Tellusim::VKCommand</b>	<b>442</b>
<b>Tellusim::Compute</b>	<b>98</b>
<b>Tellusim::D3D12Compute</b>	<b>161</b>
<b>Tellusim::FUCompute</b>	<b>230</b>
<b>Tellusim::MTLCompute</b>	<b>327</b>
<b>Tellusim::VKCompute</b>	<b>443</b>
<b>Tellusim::RadixMap&lt; Key, Type, Size &gt;::ConstIterator</b>	<b>101</b>
<b>Tellusim::Context</b>	<b>101</b>
<b>Tellusim::CUContext</b>	<b>149</b>
<b>Tellusim::D3D11Context</b>	<b>153</b>
<b>Tellusim::D3D12Context</b>	<b>162</b>
<b>Tellusim::GLContext</b>	<b>243</b>
<b>Tellusim::GLESContext</b>	<b>245</b>
<b>Tellusim::HIPContext</b>	<b>257</b>
<b>Tellusim::MTLContext</b>	<b>328</b>
<b>Tellusim::VKContext</b>	<b>444</b>
<b>Tellusim::Control</b>	<b>103</b>
<b>Tellusim::ControlArea</b>	<b>107</b>
<b>Tellusim::ControlBase</b>	<b>109</b>
<b>Tellusim::ControlGrid</b>	<b>121</b>
<b>Tellusim::ControlRect</b>	<b>126</b>
<b>Tellusim::ControlPanel</b>	<b>125</b>
<b>Tellusim::ControlDialog</b>	<b>117</b>
<b>Tellusim::ControlWindow</b>	<b>144</b>
<b>Tellusim::ControlRoot</b>	<b>129</b>
<b>Tellusim::ControlSplit</b>	<b>137</b>
<b>Tellusim::ControlText</b>	<b>139</b>
<b>Tellusim::ControlButton</b>	<b>111</b>

Tellusim::ControlCheck	113
Tellusim::ControlCombo	114
Tellusim::ControlEdit	119
Tellusim::ControlGroup	122
Tellusim::ControlScroll	132
Tellusim::ControlSlider	134
Tellusim::ControlTree	141
Tellusim::CubeFilter	145
Tellusim::D3D12Tracing::D3D12Instance	164
Tellusim::Date	173
Tellusim::DecoderJPEG	174
Tellusim::DefaultDestructor< Type >	177
Tellusim::Desktop	177
Tellusim::Device	178
Tellusim::D3D11Device	154
Tellusim::D3D12Device	163
Tellusim::FUDevice	231
Tellusim::MTLDevice	329
Tellusim::VKDevice	445
Tellusim::DialogColor	184
Tellusim::DialogDirectory	185
Tellusim::DialogFileOpen	185
Tellusim::DialogFileSave	186
Tellusim::DialogMenu	187
Tellusim::DialogMessage	189
Tellusim::DialogProgress	190
Tellusim::Directory	191
Tellusim::Compute::DispatchIndirect	193
Tellusim::BitonicSort::DispatchParameters	193
Tellusim::PrefixScan::DispatchParameters	193
Tellusim::RadixSort::DispatchParameters	194
Tellusim::SpatialGrid::DispatchParameters	194

<b>Tellusim::SpatialTree::DispatchParameters</b>	<b>194</b>
<b>Tellusim::Command::DrawArraysIndirect</b>	<b>194</b>
<b>Tellusim::Command::DrawElementsIndirect</b>	<b>195</b>
<b>Tellusim::Command::DrawMeshIndirect</b>	<b>195</b>
<b>Tellusim::EncoderASTC</b>	<b>195</b>
<b>Tellusim::EncoderBC15</b>	<b>197</b>
<b>Tellusim::EncoderBC67</b>	<b>198</b>
<b>Tellusim::f32u32</b>	<b>199</b>
<b>Tellusim::f64u64</b>	<b>200</b>
<b>Tellusim::Face</b>	<b>200</b>
<b>Tellusim::BrepModel::FaceParameters</b>	<b>201</b>
<b>Tellusim::Device::Features</b>	<b>202</b>
<b>Tellusim::Fence</b>	<b>204</b>
<b>Tellusim::FUFence</b>	<b>232</b>
<b>Tellusim::VKFence</b>	<b>447</b>
<b>Tellusim::FixedPool&lt; Type, Index &gt;</b>	<b>206</b>
<b>Tellusim::float16_t</b>	<b>207</b>
<b>Tellusim::float16x4_t</b>	<b>209</b>
<b>Tellusim::float16x8_t</b>	<b>210</b>
<b>Tellusim::float21_t</b>	<b>212</b>
<b>Tellusim::float24_t</b>	<b>213</b>
<b>Tellusim::float32x4_t</b>	<b>214</b>
<b>Tellusim::float32x8_t</b>	<b>216</b>
<b>Tellusim::float64x2_t</b>	<b>217</b>
<b>Tellusim::float64x4_t</b>	<b>218</b>
<b>Tellusim::float64x8_t</b>	<b>220</b>
<b>Tellusim::Font</b>	<b>221</b>
<b>Tellusim::FontBatch</b>	<b>223</b>
<b>Tellusim::FontBatch32</b>	<b>224</b>
<b>Tellusim::FontStyle</b>	<b>225</b>
<b>Tellusim::FourierTransform</b>	<b>226</b>
<b>Tellusim::GradientStyle</b>	<b>254</b>

<b>Tellusim::HeapPool&lt; Threshold &gt;</b>	<b>255</b>
<b>Tellusim::Image</b>	<b>259</b>
<b>Tellusim::ImageColor</b>	<b>264</b>
<b>Tellusim::ImageSampler</b>	<b>265</b>
<b>Tellusim::ImageStream</b>	<b>267</b>
<b>Tellusim::Info</b>	<b>268</b>
<b>Tellusim::Tracing::Instance</b>	<b>269</b>
<b>Tellusim::int16x8_t</b>	<b>269</b>
<b>Tellusim::int32x4_t</b>	<b>271</b>
<b>Tellusim::int32x8_t</b>	<b>273</b>
<b>Tellusim::IsPtr&lt; Type &gt;</b>	<b>275</b>
<b>Tellusim::RadixMap&lt; Key, Type, Size &gt;::Iterator</b>	<b>276</b>
<b>Tellusim::Json</b>	<b>276</b>
<b>Tellusim::Kernel</b>	<b>279</b>
<b>Tellusim::D3D12Kernel</b>	<b>164</b>
<b>Tellusim::FUKernel</b>	<b>233</b>
<b>Tellusim::MTLKernel</b>	<b>330</b>
<b>Tellusim::MeshTransform::KeyData&lt; Type &gt;</b>	<b>281</b>
<b>Tellusim::Layer</b>	<b>282</b>
<b>Tellusim::Matrix3x2&lt; T &gt;</b>	<b>282</b>
<b>Tellusim::Matrix4x3&lt; T &gt;</b>	<b>285</b>
<b>Tellusim::Matrix4x3&lt; float32_t &gt;</b>	<b>285</b>
<b>Tellusim::Matrix4x4&lt; T &gt;</b>	<b>289</b>
<b>Tellusim::MatrixNxM&lt; Type, N, M &gt;</b>	<b>293</b>
<b>Tellusim::Mesh</b>	<b>295</b>
<b>Tellusim::MeshAnimation</b>	<b>300</b>
<b>Tellusim::MeshAttachment</b>	<b>302</b>
<b>Tellusim::MeshAttribute</b>	<b>304</b>
<b>Tellusim::MeshGeometry</b>	<b>306</b>
<b>Tellusim::MeshIndices</b>	<b>312</b>
<b>Tellusim::MeshJoint</b>	<b>314</b>
<b>Tellusim::MeshModel::Meshlet</b>	<b>315</b>

<b>Tellusim::MeshMaterial</b>	<b>315</b>
<b>Tellusim::MeshModel</b>	<b>317</b>
<b>Tellusim::MeshNode</b>	<b>321</b>
<b>Tellusim::MeshStream</b>	<b>322</b>
<b>Tellusim::MeshTransform</b>	<b>323</b>
<b>Tellusim::Mipmap</b>	<b>324</b>
<b>Tellusim::MTLTracing::MTLInstance</b>	<b>330</b>
<b>Tellusim::Spatial::Node&lt; Type, Size &gt;</b>	<b>338</b>
<b>Tellusim::Atlas&lt; Type &gt;::Node</b>	<b>339</b>
<b>Tellusim::SpatialTree::Node</b>	<b>340</b>
<b>Tellusim::Origin</b>	<b>340</b>
<b>Tellusim::Pipeline</b>	<b>340</b>
<b>Tellusim::D3D12Pipeline</b>	<b>165</b>
<b>Tellusim::FUPipeline</b>	<b>234</b>
<b>Tellusim::MTLPipeline</b>	<b>331</b>
<b>Tellusim::Polygon&lt; Capacity &gt;</b>	<b>347</b>
<b>Tellusim::PrefixScan</b>	<b>349</b>
<b>Tellusim::Quaternion&lt; T &gt;</b>	<b>353</b>
<b>Tellusim::Query</b>	<b>356</b>
<b>Tellusim::FUQuery</b>	<b>235</b>
<b>Tellusim::RadixCompare&lt; Type &gt;</b>	<b>357</b>
<b>Tellusim::RadixCompare&lt; float32_t &gt;</b>	<b>358</b>
<b>Tellusim::RadixCompare&lt; int32_t &gt;</b>	<b>358</b>
<b>Tellusim::RadixCompare&lt; uint32_t &gt;</b>	<b>358</b>
<b>Tellusim::RadixMap&lt; Key, Type, Size &gt;</b>	<b>358</b>
<b>Tellusim::RadixSort</b>	<b>359</b>
<b>Tellusim::Random&lt; Integer, Float &gt;</b>	<b>363</b>
<b>Tellusim::Rect</b>	<b>364</b>
<b>Tellusim::Region</b>	<b>366</b>
<b>Tellusim::Sampler</b>	<b>366</b>
<b>Tellusim::FUSampler</b>	<b>236</b>
<b>Tellusim::MTLSampler</b>	<b>332</b>

Tellusim::Scissor	368
Tellusim::SeparableFilter	369
Tellusim::Shader	371
Tellusim::CUShader	150
Tellusim::D3D11Shader	155
Tellusim::D3D12Shader	166
Tellusim::FUShader	237
Tellusim::GLESShader	246
Tellusim::GLShader	250
Tellusim::HIPShader	258
Tellusim::MTLShader	333
Tellusim::ShaderCompiler	375
Tellusim::VKShader	448
Tellusim::Size	376
Tellusim::Slice	377
Tellusim::SpatialGrid	382
Tellusim::SpatialTree	383
Tellusim::SpinLock	387
Tellusim::Query::Statistics	388
Tellusim::Stream	388
Tellusim::Blob	52
Tellusim::File	205
Tellusim::Socket	377
Tellusim::SocketSSL	379
Tellusim::Source	380
Tellusim::String	391
Tellusim::StrokeStyle	394
Tellusim::Surface	396
Tellusim::D3D11Surface	156
Tellusim::D3D12Surface	167
Tellusim::GLESSurface	247
Tellusim::GLSurface	251

Tellusim::MTLSurface	334
Tellusim::VKSurface	449
Tellusim::Target	397
Tellusim::D3D11Target	157
Tellusim::D3D12Target	169
Tellusim::FUTarget	238
Tellusim::GLESTarget	248
Tellusim::GLTarget	252
Tellusim::MTLTarget	335
Tellusim::VKTarget	450
Tellusim::Async::Task	399
Tellusim::Tensor	400
Tellusim::TensorGraph	402
Tellusim::Texture	405
Tellusim::CUTexture	151
Tellusim::D3D11Texture	158
Tellusim::D3D12Texture	170
Tellusim::FUTexture	239
Tellusim::GLESTexture	249
Tellusim::GLTexture	253
Tellusim::HIPTexture	259
Tellusim::MTLTexture	336
Tellusim::VKTexture	451
Tellusim::TextureTable	408
Tellusim::Thread	409
Tellusim::ThreadClassFunction0< Class, Func >	410
Tellusim::ThreadClassFunction1< Class, Func, A0 >	411
Tellusim::ThreadClassFunction2< Class, Func, A0, A1 >	412
Tellusim::ThreadClassFunction3< Class, Func, A0, A1, A2 >	413
Tellusim::ThreadClassFunction4< Class, Func, A0, A1, A2, A3 >	414
Tellusim::ThreadFunction0< Func >	415
Tellusim::ThreadFunction1< Func, A0 >	416



<b>Tellusim::ThreadFunction2&lt; Func, A0, A1 &gt;</b>	<b><a href="#">417</a></b>
<b>Tellusim::ThreadFunction3&lt; Func, A0, A1, A2 &gt;</b>	<b><a href="#">418</a></b>
<b>Tellusim::ThreadFunction4&lt; Func, A0, A1, A2, A3 &gt;</b>	<b><a href="#">419</a></b>
<b>Tellusim::Tracing</b>	<b><a href="#">420</a></b>
<b>Tellusim::D3D12Tracing</b>	<b><a href="#">171</a></b>
<b>Tellusim::FUTracing</b>	<b><a href="#">240</a></b>
<b>Tellusim::MTLTracing</b>	<b><a href="#">337</a></b>
<b>Tellusim::VKTracing</b>	<b><a href="#">452</a></b>
<b>Tellusim::Traversal</b>	<b><a href="#">423</a></b>
<b>Tellusim::D3D12Traversal</b>	<b><a href="#">172</a></b>
<b>Tellusim::FUTraversal</b>	<b><a href="#">241</a></b>
<b>Tellusim::uint16x8_t</b>	<b><a href="#">426</a></b>
<b>Tellusim::uint32x4_t</b>	<b><a href="#">427</a></b>
<b>Tellusim::uint32x8_t</b>	<b><a href="#">429</a></b>
<b>Tellusim::Vector2&lt; T &gt;</b>	<b><a href="#">431</a></b>
<b>Tellusim::Vector2&lt; float32_t &gt;</b>	<b><a href="#">431</a></b>
<b>Tellusim::Vector2&lt; uint32_t &gt;</b>	<b><a href="#">431</a></b>
<b>Tellusim::Vector3&lt; T &gt;</b>	<b><a href="#">433</a></b>
<b>Tellusim::Vector3&lt; float32_t &gt;</b>	<b><a href="#">433</a></b>
<b>Tellusim::Vector4&lt; T &gt;</b>	<b><a href="#">436</a></b>
<b>Tellusim::Vector4&lt; float32_t &gt;</b>	<b><a href="#">436</a></b>
<b>Tellusim::Vector4&lt; uint32_t &gt;</b>	<b><a href="#">436</a></b>
<b>Tellusim::VectorN&lt; Type, N &gt;</b>	<b><a href="#">439</a></b>
<b>Tellusim::Viewport</b>	<b><a href="#">440</a></b>
<b>Tellusim::VKTracing::VKInstance</b>	<b><a href="#">447</a></b>
<b>Tellusim::Window</b>	<b><a href="#">453</a></b>
<b>Tellusim::Xml</b>	<b><a href="#">459</a></b>

## 3 Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Tellusim::App	38
Tellusim::Archive	39
Tellusim::ArchiveStream	40
Tellusim::Async	40
Tellusim::AsyncFunction< Func >	42
Tellusim::AsyncFunctionBase	43
Tellusim::AsyncFunctionRet< Type >	43
Tellusim::AsyncFunctionRet< void >	44
Tellusim::Atlas< Type >	44
Tellusim::AtomicArray< Type >	45
Tellusim::Atomici32	46
Tellusim::AtomicLock	47
Tellusim::AtomicPtr< Type >	47
Tellusim::BitonicSort	48
Tellusim::Blob	52
Tellusim::BoundingBox< T, Vector3 >	53
Tellusim::BoundCircle< T, Vector2 >	56
Tellusim::BoundFrustum< T >	58
Tellusim::BoundRect< T, Vector2 >	61
Tellusim::BoundSphere< T, Vector3 >	63
Tellusim::BrepModel	66
Tellusim::Buffer	68
Tellusim::BufferTable	70
Tellusim::Tracing::BuildIndirect Build indirect parameters	71
Tellusim::Canvas	71
Tellusim::CanvasElement	74
Tellusim::CanvasEllipse	77
Tellusim::CanvasMesh	79
Tellusim::CanvasRect	81
Tellusim::CanvasShape	82
Tellusim::CanvasShapeVertex	84

<a href="#">Tellusim::CanvasStrip</a>	85
<a href="#">Tellusim::CanvasStripVertex</a>	86
<a href="#">Tellusim::CanvasText</a>	88
<a href="#">Tellusim::CanvasTriangle</a>	89
<a href="#">Tellusim::CanvasVertex</a>	91
<a href="#">Tellusim::Color</a>	92
<a href="#">Tellusim::Command</a>	94
<a href="#">Tellusim::Compute</a>	98
<a href="#">Tellusim::RadixMap&lt; Key, Type, Size &gt;::ConstIterator</a> Constant iterator	101
<a href="#">Tellusim::Context</a>	101
<a href="#">Tellusim::Control</a>	103
<a href="#">Tellusim::ControlArea</a>	107
<a href="#">Tellusim::ControlBase</a>	109
<a href="#">Tellusim::ControlButton</a>	111
<a href="#">Tellusim::ControlCheck</a>	113
<a href="#">Tellusim::ControlCombo</a>	114
<a href="#">Tellusim::ControlDialog</a>	117
<a href="#">Tellusim::ControlEdit</a>	119
<a href="#">Tellusim::ControlGrid</a>	121
<a href="#">Tellusim::ControlGroup</a>	122
<a href="#">Tellusim::ControlPanel</a>	125
<a href="#">Tellusim::ControlRect</a>	126
<a href="#">Tellusim::ControlRoot</a>	129
<a href="#">Tellusim::ControlScroll</a>	132
<a href="#">Tellusim::ControlSlider</a>	134
<a href="#">Tellusim::ControlSplit</a>	137
<a href="#">Tellusim::ControlText</a>	139
<a href="#">Tellusim::ControlTree</a>	141
<a href="#">Tellusim::ControlWindow</a>	144
<a href="#">Tellusim::CubeFilter</a>	145
<a href="#">Tellusim::CUBuffer</a>	147

<a href="#">Tellusim::CUContext</a>	<a href="#">149</a>
<a href="#">Tellusim::CUShader</a>	<a href="#">150</a>
<a href="#">Tellusim::CUTexture</a>	<a href="#">151</a>
<a href="#">Tellusim::D3D11Buffer</a>	<a href="#">152</a>
<a href="#">Tellusim::D3D11Context</a>	<a href="#">153</a>
<a href="#">Tellusim::D3D11Device</a>	<a href="#">154</a>
<a href="#">Tellusim::D3D11Shader</a>	<a href="#">155</a>
<a href="#">Tellusim::D3D11Surface</a>	<a href="#">156</a>
<a href="#">Tellusim::D3D11Target</a>	<a href="#">157</a>
<a href="#">Tellusim::D3D11Texture</a>	<a href="#">158</a>
<a href="#">Tellusim::D3D12Buffer</a>	<a href="#">159</a>
<a href="#">Tellusim::D3D12Command</a>	<a href="#">160</a>
<a href="#">Tellusim::D3D12Compute</a>	<a href="#">161</a>
<a href="#">Tellusim::D3D12Context</a>	<a href="#">162</a>
<a href="#">Tellusim::D3D12Device</a>	<a href="#">163</a>
<a href="#">Tellusim::D3D12Tracing::D3D12Instance Tracing instance</a>	<a href="#">164</a>
<a href="#">Tellusim::D3D12Kernel</a>	<a href="#">164</a>
<a href="#">Tellusim::D3D12Pipeline</a>	<a href="#">165</a>
<a href="#">Tellusim::D3D12Shader</a>	<a href="#">166</a>
<a href="#">Tellusim::D3D12Surface</a>	<a href="#">167</a>
<a href="#">Tellusim::D3D12Target</a>	<a href="#">169</a>
<a href="#">Tellusim::D3D12Texture</a>	<a href="#">170</a>
<a href="#">Tellusim::D3D12Tracing</a>	<a href="#">171</a>
<a href="#">Tellusim::D3D12Traversal</a>	<a href="#">172</a>
<a href="#">Tellusim::Date</a>	<a href="#">173</a>
<a href="#">Tellusim::DecoderJPEG</a>	<a href="#">174</a>
<a href="#">Tellusim::DefaultDestructor&lt; Type &gt;</a>	<a href="#">177</a>
<a href="#">Tellusim::Desktop</a>	<a href="#">177</a>
<a href="#">Tellusim::Device</a>	<a href="#">178</a>
<a href="#">Tellusim::DialogColor</a>	<a href="#">184</a>
<a href="#">Tellusim::DialogDirectory</a>	<a href="#">185</a>

<a href="#">Tellusim::DialogFileOpen</a>	185
<a href="#">Tellusim::DialogFileSave</a>	186
<a href="#">Tellusim::DialogMenu</a>	187
<a href="#">Tellusim::DialogMessage</a>	189
<a href="#">Tellusim::DialogProgress</a>	190
<a href="#">Tellusim::Directory</a>	191
<a href="#">Tellusim::Compute::DispatchIndirect</a> Dispatch indirect parameters	193
<a href="#">Tellusim::BitonicSort::DispatchParameters</a>	193
<a href="#">Tellusim::PrefixScan::DispatchParameters</a>	193
<a href="#">Tellusim::RadixSort::DispatchParameters</a>	194
<a href="#">Tellusim::SpatialGrid::DispatchParameters</a>	194
<a href="#">Tellusim::SpatialTree::DispatchParameters</a>	194
<a href="#">Tellusim::Command::DrawArraysIndirect</a> Draw arrays indirect parameters	194
<a href="#">Tellusim::Command::DrawElementsIndirect</a> Draw elements indirect parameters	195
<a href="#">Tellusim::Command::DrawMeshIndirect</a> Draw mesh indirect parameters	195
<a href="#">Tellusim::EncoderASTC</a>	195
<a href="#">Tellusim::EncoderBC15</a>	197
<a href="#">Tellusim::EncoderBC67</a>	198
<a href="#">Tellusim::f32u32</a>	199
<a href="#">Tellusim::f64u64</a>	200
<a href="#">Tellusim::Face</a>	200
<a href="#">Tellusim::BrepModel::FaceParameters</a>	201
<a href="#">Tellusim::Device::Features</a> Device features	202
<a href="#">Tellusim::Fence</a>	204
<a href="#">Tellusim::File</a>	205
<a href="#">Tellusim::FixedPool&lt; Type, Index &gt;</a>	206
<a href="#">Tellusim::float16_t</a>	207
<a href="#">Tellusim::float16x4_t</a>	209
<a href="#">Tellusim::float16x8_t</a>	210

<a href="#">Tellusim::float21_t</a>	<a href="#">212</a>
<a href="#">Tellusim::float24_t</a>	<a href="#">213</a>
<a href="#">Tellusim::float32x4_t</a>	<a href="#">214</a>
<a href="#">Tellusim::float32x8_t</a>	<a href="#">216</a>
<a href="#">Tellusim::float64x2_t</a>	<a href="#">217</a>
<a href="#">Tellusim::float64x4_t</a>	<a href="#">218</a>
<a href="#">Tellusim::float64x8_t</a>	<a href="#">220</a>
<a href="#">Tellusim::Font</a>	<a href="#">221</a>
<a href="#">Tellusim::FontBatch</a>	<a href="#">223</a>
<a href="#">Tellusim::FontBatch32</a>	<a href="#">224</a>
<a href="#">Tellusim::FontStyle</a>	<a href="#">225</a>
<a href="#">Tellusim::FourierTransform</a>	<a href="#">226</a>
<a href="#">Tellusim::FUBuffer</a>	<a href="#">228</a>
<a href="#">Tellusim::FUCommand</a>	<a href="#">229</a>
<a href="#">Tellusim::FUCompute</a>	<a href="#">230</a>
<a href="#">Tellusim::FUDevice</a>	<a href="#">231</a>
<a href="#">Tellusim::FUFence</a>	<a href="#">232</a>
<a href="#">Tellusim::FUKernel</a>	<a href="#">233</a>
<a href="#">Tellusim::FUPipeline</a>	<a href="#">234</a>
<a href="#">Tellusim::FUQuery</a>	<a href="#">235</a>
<a href="#">Tellusim::FUSampler</a>	<a href="#">236</a>
<a href="#">Tellusim::FUShader</a>	<a href="#">237</a>
<a href="#">Tellusim::FUTarget</a>	<a href="#">238</a>
<a href="#">Tellusim::FUTexture</a>	<a href="#">239</a>
<a href="#">Tellusim::FUTracing</a>	<a href="#">240</a>
<a href="#">Tellusim::FUTraversal</a>	<a href="#">241</a>
<a href="#">Tellusim::GLBuffer</a>	<a href="#">242</a>
<a href="#">Tellusim::GLContext</a>	<a href="#">243</a>
<a href="#">Tellusim::GLESBuffer</a>	<a href="#">244</a>
<a href="#">Tellusim::GLESTexture</a>	<a href="#">245</a>
<a href="#">Tellusim::GLESShader</a>	<a href="#">246</a>
<a href="#">Tellusim::GLESSurface</a>	<a href="#">247</a>

<a href="#">Tellusim::GLESTarget</a>	248
<a href="#">Tellusim::GLESTexture</a>	249
<a href="#">Tellusim::GLShader</a>	250
<a href="#">Tellusim::GLSurface</a>	251
<a href="#">Tellusim::GLTarget</a>	252
<a href="#">Tellusim::GLTexture</a>	253
<a href="#">Tellusim::GradientStyle</a>	254
<a href="#">Tellusim::HeapPool&lt; Threshold &gt;</a>	255
<a href="#">Tellusim::HIPBuffer</a>	256
<a href="#">Tellusim::HIPContext</a>	257
<a href="#">Tellusim::HIPShader</a>	258
<a href="#">Tellusim::HIPTexture</a>	259
<a href="#">Tellusim::Image</a>	259
<a href="#">Tellusim::ImageColor</a>	264
<a href="#">Tellusim::ImageSampler</a>	265
<a href="#">Tellusim::ImageStream</a>	267
<a href="#">Tellusim::Info</a>	268
<a href="#">Tellusim::Tracing::Instance</a> <a href="#">Tracing instance</a>	269
<a href="#">Tellusim::int16x8_t</a>	269
<a href="#">Tellusim::int32x4_t</a>	271
<a href="#">Tellusim::int32x8_t</a>	273
<a href="#">Tellusim::IsPtr&lt; Type &gt;</a>	275
<a href="#">Tellusim::RadixMap&lt; Key, Type, Size &gt;::Iterator</a> <a href="#">Iterator</a>	276
<a href="#">Tellusim::Json</a>	276
<a href="#">Tellusim::Kernel</a>	279
<a href="#">Tellusim::MeshTransform::KeyData&lt; Type &gt;</a> <a href="#">Transform data</a>	281
<a href="#">Tellusim::Layer</a>	282
<a href="#">Tellusim::Matrix3x2&lt; T &gt;</a>	282
<a href="#">Tellusim::Matrix4x3&lt; T &gt;</a>	285
<a href="#">Tellusim::Matrix4x4&lt; T &gt;</a>	289

Tellusim::MatrixNxM< Type, N, M >	293
Tellusim::Mesh	295
Tellusim::MeshAnimation	300
Tellusim::MeshAttachment	302
Tellusim::MeshAttribute	304
Tellusim::MeshGeometry	306
Tellusim::MeshIndices	312
Tellusim::MeshJoint	314
Tellusim::MeshModel::Meshlet	315
Tellusim::MeshMaterial	315
Tellusim::MeshModel	317
Tellusim::MeshNode	321
Tellusim::MeshStream	322
Tellusim::MeshTransform	323
Tellusim::Mipmap	324
Tellusim::MTLBuffer	325
Tellusim::MTLCommand	326
Tellusim::MTLCompute	327
Tellusim::MTLContext	328
Tellusim::MTLDevice	329
Tellusim::MTLTracing::MTLInstance Tracing instance	330
Tellusim::MTLKernel	330
Tellusim::MTLPipeline	331
Tellusim::MTLSampler	332
Tellusim::MTLShader	333
Tellusim::MTLSurface	334
Tellusim::MTLTarget	335
Tellusim::MTLTexture	336
Tellusim::MTLTracing	337
Tellusim::Spatial::Node< Type, Size > Spatial Node	338



Tellusim::Atlas< Type >::Node Atlas Node	339
Tellusim::SpatialTree::Node	340
Tellusim::Origin	340
Tellusim::Pipeline	340
Tellusim::Polygon< Capacity >	347
Tellusim::PrefixScan	349
Tellusim::Quaternion< T >	353
Tellusim::Query	356
Tellusim::RadixCompare< Type >	357
Tellusim::RadixCompare< float32_t >	358
Tellusim::RadixCompare< int32_t >	358
Tellusim::RadixCompare< uint32_t >	358
Tellusim::RadixMap< Key, Type, Size >	358
Tellusim::RadixSort	359
Tellusim::Random< Integer, Float >	363
Tellusim::Rect	364
Tellusim::Region	366
Tellusim::Sampler	366
Tellusim::Scissor	368
Tellusim::SeparableFilter	369
Tellusim::Shader	371
Tellusim::ShaderCompiler	375
Tellusim::Size	376
Tellusim::Slice	377
Tellusim::Socket	377
Tellusim::SocketSSL	379
Tellusim::Source	380
Tellusim::SpatialGrid	382
Tellusim::SpatialTree	383
Tellusim::SpinLock	387
Tellusim::Query::Statistics Statistics query	388

<a href="#">Tellusim::Stream</a>	<a href="#">388</a>
<a href="#">Tellusim::String</a>	<a href="#">391</a>
<a href="#">Tellusim::StrokeStyle</a>	<a href="#">394</a>
<a href="#">Tellusim::Surface</a>	<a href="#">396</a>
<a href="#">Tellusim::Target</a>	<a href="#">397</a>
<a href="#">Tellusim::Async::Task</a> <a href="#">Task</a>	<a href="#">399</a>
<a href="#">Tellusim::Tensor</a>	<a href="#">400</a>
<a href="#">Tellusim::TensorGraph</a>	<a href="#">402</a>
<a href="#">Tellusim::Texture</a>	<a href="#">405</a>
<a href="#">Tellusim::TextureTable</a>	<a href="#">408</a>
<a href="#">Tellusim::Thread</a>	<a href="#">409</a>
<a href="#">Tellusim::ThreadClassFunction0&lt; Class, Func &gt;</a>	<a href="#">410</a>
<a href="#">Tellusim::ThreadClassFunction1&lt; Class, Func, A0 &gt;</a>	<a href="#">411</a>
<a href="#">Tellusim::ThreadClassFunction2&lt; Class, Func, A0, A1 &gt;</a>	<a href="#">412</a>
<a href="#">Tellusim::ThreadClassFunction3&lt; Class, Func, A0, A1, A2 &gt;</a>	<a href="#">413</a>
<a href="#">Tellusim::ThreadClassFunction4&lt; Class, Func, A0, A1, A2, A3 &gt;</a>	<a href="#">414</a>
<a href="#">Tellusim::ThreadFunction0&lt; Func &gt;</a>	<a href="#">415</a>
<a href="#">Tellusim::ThreadFunction1&lt; Func, A0 &gt;</a>	<a href="#">416</a>
<a href="#">Tellusim::ThreadFunction2&lt; Func, A0, A1 &gt;</a>	<a href="#">417</a>
<a href="#">Tellusim::ThreadFunction3&lt; Func, A0, A1, A2 &gt;</a>	<a href="#">418</a>
<a href="#">Tellusim::ThreadFunction4&lt; Func, A0, A1, A2, A3 &gt;</a>	<a href="#">419</a>
<a href="#">Tellusim::Tracing</a>	<a href="#">420</a>
<a href="#">Tellusim::Traversal</a>	<a href="#">423</a>
<a href="#">Tellusim::uint16x8_t</a>	<a href="#">426</a>
<a href="#">Tellusim::uint32x4_t</a>	<a href="#">427</a>
<a href="#">Tellusim::uint32x8_t</a>	<a href="#">429</a>
<a href="#">Tellusim::Vector2&lt; T &gt;</a>	<a href="#">431</a>
<a href="#">Tellusim::Vector3&lt; T &gt;</a>	<a href="#">433</a>
<a href="#">Tellusim::Vector4&lt; T &gt;</a>	<a href="#">436</a>
<a href="#">Tellusim::VectorN&lt; Type, N &gt;</a>	<a href="#">439</a>
<a href="#">Tellusim::Viewport</a>	<a href="#">440</a>

<a href="#">Tellusim::VKBuffer</a>	441
<a href="#">Tellusim::VKCommand</a>	442
<a href="#">Tellusim::VKCompute</a>	443
<a href="#">Tellusim::VKContext</a>	444
<a href="#">Tellusim::VKDevice</a>	445
<a href="#">Tellusim::VKFence</a>	447
<a href="#">Tellusim::VKTracing::VKInstance Tracing instance</a>	447
<a href="#">Tellusim::VKShader</a>	448
<a href="#">Tellusim::VKSurface</a>	449
<a href="#">Tellusim::VKTarget</a>	450
<a href="#">Tellusim::VKTexture</a>	451
<a href="#">Tellusim::VKTracing</a>	452
<a href="#">Tellusim::Window</a>	453
<a href="#">Tellusim::Xml</a>	459

## 4 Namespace Documentation

### 4.1 Tellusim::Allocator Namespace Reference

#### Functions

- void \* [allocate](#) (size\_t size)  
*raw allocation*
- void \* **reallocate** (void \*ptr, size\_t old\_size, size\_t new\_size)
- void **free** (const void \*ptr, size\_t size)
- size\_t [getMemory](#) ()  
*memory statistics*
- size\_t **getAllocations** ()

#### 4.1.1 Detailed Description

Memory allocator

#### 4.1.2 Function Documentation

#### 4.1.2.1 allocate()

```
static Type * Tellusim::Allocator::allocate (
    size_t size )
```

raw allocation

array allocation

## 4.2 Tellusim::Android Namespace Reference

### Typedefs

- using [Main](#) = int32\_t(int32\_t argc, char \*\*argv)  
*Android native activity.*

### Functions

- ANativeActivity \* [getActivity](#) ()  
*Android activity.*
- bool [isCreated](#) ()
- bool [isResumed](#) ()
- bool [isFocused](#) ()
- ANativeWindow \* [getWindow](#) ()  
*Android window.*
- int32\_t [getWidth](#) ()
- int32\_t [getHeight](#) ()
- int32\_t [getFormat](#) ()
- [String](#) [getPackageName](#) ()  
*application package name*
- [String](#) [getHomeDirectory](#) ()  
*application directories*
- [String](#) [getFilesDirectory](#) ()
- [String](#) [getCacheDirectory](#) ()
- [String](#) [getCardDirectory](#) ()
- [String](#) [getObbDirectory](#) ()
- void [onCreate](#) (ANativeActivity \*activity, void \*state, size\_t size, [Main](#) \*main)

#### 4.2.1 Detailed Description

[Android](#)

## 4.3 Tellusim::Basis Namespace Reference

#### 4.3.1 Detailed Description

[Basis](#) utils

## 4.4 Tellusim::Emscripten Namespace Reference

### Typedefs

- using [LoadedCallback](#) = Function< void([Blob](#) blob)>  
*load user file*
- using [ProgressCallback](#) = Function< void(uint32\_t progress, uint32\_t total)>  
*asynchronous fetch*
- using **FetchCallback** = Function< void(const uint8\_t \*data, size\_t size, uint32\_t status)>

### Functions

- void [run](#) (const char \*src)  
*run script*
- int32\_t **runi32** (const char \*src)
- const char \* **runs** (const char \*src)
- void [alert](#) (const char \*message)  
*alert dialog*
- void **alert** (const [String](#) &message)
- void **alertf** (const char \*format,...) 1(1)
- void void [save](#) (const [Blob](#) &blob, const char \*mime=nullptr)  
*save user file*
- void **load** (const [LoadedCallback](#) &func, const char \*type=nullptr)
- void **fetch** (const char \*name, const [FetchCallback](#) &fetch\_func, const [ProgressCallback](#) &progress\_↔  
func, bool cache=false)
- void **fetch** (const [String](#) &name, const [FetchCallback](#) &fetch\_func, const [ProgressCallback](#) &progress\_↔  
\_func, bool cache=false)

#### 4.4.1 Detailed Description

[Emscripten](#) application

## 4.5 Tellusim::Expression Namespace Reference

### Functions

- int64\_t [getScalari64](#) (const char \*src)  
*scalar expressions*
- uint64\_t **getScalaru64** (const char \*src)
- float32\_t **getScalarf32** (const char \*src)
- float64\_t **getScalarf64** (const char \*src)
- [Vector2f](#) [getVector2f](#) (const char \*src, const char \*type="Vector2f")  
*vector expressions*
- [Vector3f](#) **getVector3f** (const char \*src, const char \*type="Vector3f")
- [Vector4f](#) **getVector4f** (const char \*src, const char \*type="Vector4f")
- [Matrix3x2f](#) [getMatrix3x2f](#) (const char \*src)  
*matrix expressions*
- [Matrix4x4f](#) **getMatrix4x4f** (const char \*src)

#### 4.5.1 Detailed Description

[Expression](#) utils

### 4.6 Tellusim::iOS Namespace Reference

#### Enumerations

- enum [Orientation](#) {  
**OrientationUnknown** = 0,  
**OrientationPortrait**,  
**OrientationPortraitUpsideDown**,  
**OrientationLandscapeLeft**,  
**OrientationLandscapeRight**,  
**OrientationFaceUp**,  
**OrientationFaceDown** }

*Device orientation.*

#### Functions

- void \* [getApplication](#) ()  
*iOS application*
- bool **isCreated** ()
- bool **isFocused** ()
- uint32\_t [getWidth](#) ()  
*screen size*
- uint32\_t **getHeight** ()
- float32\_t **getScale** ()
- [String](#) [getModel](#) ()  
*device info*
- [Orientation](#) **getOrientation** ()
- bool [setKeyboardHidden](#) (bool hidden)  
*virtual keyboard*
- bool **isKeyboardHidden** ()
- [String](#) [getHomeDirectory](#) ()  
*application home directory*
- [String](#) [getDocumentsDirectory](#) ()  
*application documents directory*
- bool [openUrl](#) (const char \*name)  
*open url*
- bool **openUrl** (const [String](#) &name)

#### 4.6.1 Detailed Description

[iOS](#)

## 4.7 Tellusim::Log Namespace Reference

### Typedefs

- using [Callback](#) = bool([Level](#), uint64\_t time, const char \*str, void \*data)  
*print callback*

### Enumerations

- enum [Level](#) {  
    **Fatal** = 0,  
    **Error**,  
    **Warning**,  
    **Message**,  
    **Verbose**,  
    **Unknown**,  
    **NumLevels** }  
    *Log level.*

### Functions

- void [setLevel](#) ([Level](#) level)  
    *current Log level*
- [Level](#) [getLevel](#) ()
- void [setCallback](#) ([Callback](#) \*callback, void \*data=NULLPTR)
- [Callback](#) \* [getCallback](#) ()
- void \* [getCallbackData](#) ()
- void [unlockCallback](#) ()
- void [lockCallback](#) ()
- void [print](#) (const char \*str)  
    *print message*
- void [vprintf](#) (const char \*str, va\_list args)
- void [printf](#) (const char \*format,...) 1(1)
- void void [print](#) ([Level](#) level, const char \*str)  
    *print message with Level*
- void [prnte](#) ([Level](#) level, const char \*str)
- void [vprintf](#) ([Level](#) level, const char \*str, va\_list args)
- void [vprintf](#) ([Level](#) level, const char \*str, va\_list args)
- void [printf](#) ([Level](#) level, const char \*format,...) 1(2)
- void void [printf](#) ([Level](#) level, const char \*format,...) 1(2)

#### 4.7.1 Detailed Description

[Log](#) system

## 4.8 Tellusim::LU Namespace Reference

#### 4.8.1 Detailed Description

Lower Upper Decomposition

## 4.9 Tellusim::MeshGraph Namespace Reference

### Typedefs

- using [ProgressCallback](#) = Function< bool(uint32\_t progress)>  
*progress callback*

### Functions

- bool [create](#) ([Mesh](#) &dest, [Mesh](#) &src, uint32\_t max\_attributes, uint32\_t max\_primitives, const [ProgressCallback](#) \*func=NULLPTR, [Async](#) \*async=NULLPTR)
- bool [create](#) (Array< [MeshGeometry](#) > &dest, [MeshGeometry](#) &src, uint32\_t max\_attributes, uint32\_t max\_primitives, const [ProgressCallback](#) \*func=NULLPTR, [Async](#) \*async=NULLPTR)

### 4.9.1 Detailed Description

[MeshGraph](#) algorithms

### 4.9.2 Function Documentation

#### 4.9.2.1 create()

```
bool Tellusim::MeshGraph::create (
    Mesh & dest,
    Mesh & src,
    uint32_t max_attributes,
    uint32_t max_primitives,
    const ProgressCallback * func = NULLPTR,
    Async * async = NULLPTR )
```

mesh graph reduction

#### Parameters

<i>max_attributes</i>	Maximum number of attributes per geometry
<i>max_primitives</i>	Maximum number of primitives per geometry

## 4.10 Tellusim::MeshReduce Namespace Reference

### Typedefs

- using [ProgressCallback](#) = Function< bool(uint32\_t progress)>  
*progress callback*



## Functions

- bool [collapse](#) ([Mesh](#) &dest, const [Mesh](#) &src, float32\_t ratio, float32\_t threshold=0.0f, const ProgressCallback \*func=nullptr)
- bool **collapse** ([MeshGeometry](#) &dest, const [MeshGeometry](#) &src, float32\_t ratio, float32\_t threshold=0.0f, const ProgressCallback \*func=nullptr, uint32\_t position=Maxu32)

## 4.10.1 Detailed Description

[MeshReduce](#) algorithms

## 4.10.2 Function Documentation

## 4.10.2.1 collapse()

```
bool Tellusim::MeshReduce::collapse (
    Mesh & dest,
    const Mesh & src,
    float32_t ratio,
    float32_t threshold = 0.0f,
    const ProgressCallback * func = nullptr )
```

mesh collapse reduction

## Parameters

<i>ratio</i>	<a href="#">Triangle</a> reduction ratio (use negative ratio for border reduction)
<i>threshold</i>	Edge collapse threshold
<i>position</i>	Position attribute index

## 4.11 Tellusim::MeshRefine Namespace Reference

## Functions

- bool [subdiv](#) ([Mesh](#) &dest, const [Mesh](#) &src, uint32\_t steps)
- bool **subdiv** ([MeshGeometry](#) &dest, const [MeshGeometry](#) &src, uint32\_t steps, uint32\_t position=Maxu32)

## 4.11.1 Detailed Description

[MeshRefine](#) algorithms

## 4.11.2 Function Documentation

#### 4.11.2.1 subdiv()

```
bool Tellusim::MeshRefine::subdiv (
    Mesh & dest,
    const Mesh & src,
    uint32_t steps )
```

mesh subdivision algorithm

##### Parameters

<i>steps</i>	Number of subdivision steps
--------------	-----------------------------

## 4.12 Tellusim::MeshSolid Namespace Reference

### Typedefs

- using [ProgressCallback](#) = Function< bool(uint32\_t progress)>  
*progress callback*

### Functions

- bool [create](#) (Mesh &dest, const Mesh &src, float32\_t ratio=1.0f, float32\_t threshold=0.9f, const ProgressCallback \*func=nullptr)
- bool [create](#) (MeshGeometry &dest, const MeshGeometry &src, float32\_t ratio=1.0f, float32\_t threshold=0.9f, const ProgressCallback \*func=nullptr, uint32\_t position=Maxu32)

#### 4.12.1 Detailed Description

[MeshSolid](#) algorithms

#### 4.12.2 Function Documentation

##### 4.12.2.1 create()

```
bool Tellusim::MeshSolid::create (
    Mesh & dest,
    const Mesh & src,
    float32_t ratio = 1.0f,
    float32_t threshold = 0.9f,
    const ProgressCallback * func = nullptr )
```

advancing front generation algorithm

## Parameters

<i>ratio</i>	Tetrahedron height ratio
<i>threshold</i>	Delaunay radius threshold
<i>position</i>	Position attribute index

## 4.13 Tellusim::Order Namespace Reference

## Functions

- template<class Type = uint32\_t>  
Type [hilbert2](#) (uint32\_t size, uint32\_t x, uint32\_t y)  
*Hilbert curve.*
- template<class Type = uint32\_t>  
void [ihilbert2](#) (uint32\_t size, Type index, uint32\_t &x, uint32\_t &y)  
*inverse Hilbert curve*

## 4.13.1 Detailed Description

[Order](#) utils

## 4.14 Tellusim::Parser Namespace Reference

## Functions

- bool [isSpace](#) (char c)  
*character types*
- bool **isAlpha** (char c)
- bool **isLower** (char c)
- bool **isUpper** (char c)
- bool **isLiteral** (char c)
- bool **isDecimal** (char c)
- bool **isHexadecimal** (char c)
- bool [isComment](#) (const char \*str)  
*string types*
- bool **isNumber** (const char \*str)
- bool **isFloat** (const char \*str)
- bool **isSigned** (const char \*str)
- bool **isUnsigned** (const char \*str)
- bool **isBom** (const char \*str)
- uint32\_t [skipSpaces](#) (const char \*str)  
*skip symbols*
- uint32\_t **skipSpaces** (const char \*str, uint32\_t &line)
- uint32\_t [skipComment](#) (const char \*str)  
*skip comment*
- uint32\_t **skipComment** (const char \*str, uint32\_t &line)
- char [getSymbol](#) (const char \*str)  
*expect symbols*

- `uint32_t expectSymbol` (const char \*str, char c)
- `uint32_t expectSymbol` (const char \*str, char c, uint32\_t &line)
- `uint32_t skipToken` (const char \*str)
- `uint32_t skipToken` (const char \*str, const char \*term)
- `uint32_t readToken` (const char \*str, [String](#) &dest, bool append=false)
- `uint32_t readToken` (const char \*str, [String](#) &dest, const char \*term, bool append=false)
- `uint32_t skipName` (const char \*str)
- `uint32_t skipName` (const char \*str, const char \*pass)
- `uint32_t readName` (const char \*str, [String](#) &dest, bool append=false)
- `uint32_t readName` (const char \*str, [String](#) &dest, const char \*pass, bool append=false)
- `uint32_t skipFloat` (const char \*str)
- `uint32_t readFloat` (const char \*str, [String](#) &dest, bool append=false)
- `uint32_t skipDecimal` (const char \*str)
- `uint32_t readDecimal` (const char \*str, [String](#) &dest, bool append=false)
- `uint32_t skipHexadecimal` (const char \*str)
- `uint32_t readHexadecimal` (const char \*str, [String](#) &dest, bool append=false)
- `uint32_t skipNumber` (const char \*str)
- `uint32_t readNumber` (const char \*str, [String](#) &dest, bool append=false)
- `uint32_t skipSymbol` (const char \*str)
- `uint32_t readSymbol` (const char \*str, [String](#) &dest, bool enclose=false, bool append=false)
- `uint32_t skipString` (const char \*str)
- `uint32_t readString` (const char \*str, [String](#) &dest, bool enclose=false, bool append=false)
- `uint32_t readRegion` (const char \*str, [String](#) &dest, char from, char to, bool enclose=false, bool append=false)
- `uint32_t skipLines` (const char \*str, uint32\_t lines)
- `uint32_t skipLine` (const char \*str, bool escape=false)
- `uint32_t readLine` (const char \*str, [String](#) &dest, bool escape=false, bool append=false)
- `uint32_t readBom` (const char \*str, [String](#) &dest, bool append=false)
- *read string with byte ordered mark*
- `void error` (const char \*format,...) 1(1  
*throw [Parser](#) error*

#### 4.14.1 Detailed Description

Ascii [Parser](#)

#### 4.14.2 Function Documentation

##### 4.14.2.1 skipToken()

```
uint32_t Tellusim::Parser::skipToken (
    const char * str )
```

read space-separated token

##### Parameters

<i>append</i>	Append to the destination
---------------	---------------------------

#### 4.14.2.2 skipName()

```
uint32_t Tellusim::Parser::skipName (  
    const char * str )
```

read literal name

##### Parameters

<i>append</i>	Append to the destination
---------------	---------------------------

#### 4.14.2.3 skipFloat()

```
uint32_t Tellusim::Parser::skipFloat (  
    const char * str )
```

read floating-point number

##### Parameters

<i>append</i>	Append to the destination
---------------	---------------------------

#### 4.14.2.4 skipDecimal()

```
uint32_t Tellusim::Parser::skipDecimal (  
    const char * str )
```

read decimal number

##### Parameters

<i>append</i>	Append to the destination
---------------	---------------------------

#### 4.14.2.5 skipHexadecimal()

```
uint32_t Tellusim::Parser::skipHexadecimal (  
    const char * str )
```

read hexadecimal number

##### Parameters

<i>append</i>	Append to the destination
---------------	---------------------------

#### 4.14.2.6 skipNumber()

```
uint32_t Tellusim::Parser::skipNumber (
    const char * str )
```

read floating-point or integer number

##### Parameters

<i>append</i>	Append to the destination
---------------	---------------------------

#### 4.14.2.7 skipSymbol()

```
uint32_t Tellusim::Parser::skipSymbol (
    const char * str )
```

read single-quoted symbol

##### Parameters

<i>enclose</i>	Include quote symbols
<i>append</i>	Append to the destination

#### 4.14.2.8 skipString()

```
uint32_t Tellusim::Parser::skipString (
    const char * str )
```

read quoted string

##### Parameters

<i>enclose</i>	Include quote symbols
<i>append</i>	Append to the destination

#### 4.14.2.9 readRegion()

```
uint32_t Tellusim::Parser::readRegion (
    const char * str,
    String & dest,
    char from,
    char to,
```

```
bool enclose = false,
bool append = false )
```

read region of symbols

#### Parameters

<i>enclose</i>	Include from/begin symbols
<i>append</i>	Append to the destination

#### 4.14.2.10 skipLines()

```
uint32_t Tellusim::Parser::skipLines (
    const char * str,
    uint32_t lines )
```

read line of symbols

#### Parameters

<i>escape</i>	Read multiple lines
<i>append</i>	Append to the destination

## 4.15 Tellusim::QR Namespace Reference

### 4.15.1 Detailed Description

Orthogonal Decomposition

## 4.16 Tellusim::Quadrilateral Namespace Reference

### 4.16.1 Detailed Description

[Quadrilateral](#) utils

## 4.17 Tellusim::Spatial Namespace Reference

#### Classes

- struct [Node](#)  
*Spatial Node.*

## Typedefs

- using **Node2f** = **Node**< **BoundRect**< float32\_t, **Vector2**< float32\_t > >, 2 >  
*Node types.*
- using **Node2d** = **Node**< **BoundRect**< float64\_t, **Vector2**< float64\_t > >, 2 >
- using **Node3f** = **Node**< **BoundingBox**< float32\_t, **Vector3**< float32\_t > >, 3 >
- using **Node3d** = **Node**< **BoundingBox**< float64\_t, **Vector3**< float64\_t > >, 3 >
- using **Node4f** = **Node**< **BoundingBox**< float32\_t, **Vector4**< float32\_t > >, 3 >
- using **Node4d** = **Node**< **BoundingBox**< float64\_t, **Vector4**< float64\_t > >, 3 >

## Enumerations

- enum {  
    **MinIterations** = 1,  
    **MaxIterations** = 8,  
    **DefaultIterations** = 4 }  
    *number of iterations*

## Functions

- template<class Type , class Vector >  
    Type **get\_weight** (const **BoundRect**< Type, Vector > &bound)  
    *Create spatial tree.*
- template<class Type , class Vector >  
    Type **get\_weight** (const **BoundingBox**< Type, Vector > &bound)
- template<class Bound >  
    Bound::Type **get\_weight** (const Bound &bound)

### 4.17.1 Detailed Description

## Spatial utils

## 4.18 Tellusim::SVD Namespace Reference

## Functions

- template<class Type >  
    Type **pythagorean** (Type a, Type b)  
    *Computes  $(a^2 + b^2)^{1/2}$  function.*

### 4.18.1 Detailed Description

## Singular Value Decomposition



## 4.19 Tellusim::System Namespace Reference

### Functions

- `uint32_t` [getThreadID](#) ()
- `bool` [setEnvironment](#) (const char \*name, const char \*value)
- `bool` [setEnvironment](#) (const [String](#) &name, const char \*value)
- [String](#) [getEnvironment](#) (const char \*name)
- [String](#) [getEnvironment](#) (const [String](#) &name)
- `void *` [loadLibrary](#) (const char \*name)
- `void *` [loadLibrary](#) (const [String](#) &name)
- `void *` [getFunction](#) (void \*handle, const char \*name)
- `void *` [getFunction](#) (void \*handle, const [String](#) &name)
- `void` [closeLibrary](#) (void \*handle)
- `bool` [exec](#) (const char \*command, bool wait=false)
- `bool` [exec](#) (const [String](#) &command, bool wait=false)
- `bool` [open](#) (const char \*command)
- `bool` [open](#) (const [String](#) &command)

### 4.19.1 Detailed Description

[System](#) utils

### 4.19.2 Function Documentation

#### 4.19.2.1 [getThreadID](#)()

```
uint32_t Tellusim::System::getThreadID ( )
```

[Thread](#) identifier

#### 4.19.2.2 [setEnvironment](#)()

```
bool Tellusim::System::setEnvironment (
    const char * name,
    const char * value )
```

Environment variables

#### 4.19.2.3 [loadLibrary](#)()

```
void* Tellusim::System::loadLibrary (
    const char * name )
```

Dynamic libraries

#### 4.19.2.4 `exec()`

```
bool Tellusim::System::exec (
    const char * command,
    bool wait = false )
```

Execute command

#### 4.19.2.5 `open()`

```
bool Tellusim::System::open (
    const char * command )
```

Open resource

### 4.20 Tellusim::Time Namespace Reference

#### Enumerations

- enum {  
**Seconds** = 1000000u,  
**MSeconds** = 1000u,  
**USeconds** = 1u }

#### Functions

- uint64\_t [current](#) ()  
*current system time in microseconds*
- float64\_t [seconds](#) ()  
*current process time in seconds*
- void [sleep](#) (uint32\_t usec)  
*sleep process in microseconds*

#### 4.20.1 Detailed Description

[Time](#) utils

### 4.21 Tellusim::Triangle Namespace Reference

#### 4.21.1 Detailed Description

[Triangle](#) utils

### 4.22 Tellusim::WinApp Namespace Reference

#### Typedefs

- using [Main](#) = int32\_t(int32\_t argc, char \*\*argv)  
[WinApp](#) main.

## Functions

- void \* [getInstance](#) ()  
*application instance*
- int32\_t [getShowMode](#) ()  
*application show mode*
- void \* [getWindow](#) ()  
*application window*
- [String](#) [getLocalDirectory](#) ()  
*application directories*
- [String](#) [getCacheDirectory](#) ()
- [String](#) [getTempDirectory](#) ()
- void [main](#) (void \*instance, void \*prev\_instance, wchar\_t \*command, int32\_t show\_mode, [Main](#) \*main)

## 4.22.1 Detailed Description

[WinApp](#) application

## 4.23 Tellusim::Windows Namespace Reference

## Typedefs

- using [Main](#) = int32\_t(int32\_t argc, char \*\*argv)  
*Windows main.*

## Functions

- void \* [getInstance](#) ()  
*application instance*
- int32\_t [getShowMode](#) ()  
*application show mode*
- void \* [getConsoleHandle](#) ()  
*console window handle*
- bool [isConsoleCreated](#) ()  
*check console status*
- bool [createConsole](#) (const char \*title, uint32\_t width=0, uint32\_t height=0, int32\_t x=Maxi32, int32\_t y=Maxi32)  
*create console window*
- bool [createConsole](#) (const [String](#) &title, uint32\_t width=0, uint32\_t height=0, int32\_t x=Maxi32, int32\_t y=Maxi32)
- void [setConsoleTitle](#) (const char \*title)  
*set console title*
- void [setConsoleTitle](#) (const [String](#) &title)
- [String](#) [getConsoleTitle](#) ()
- void [setConsoleGeometry](#) (uint32\_t width, uint32\_t height, int32\_t x=Maxi32, int32\_t y=Maxi32)  
*console geometry*
- uint32\_t [getConsoleWidth](#) ()
- uint32\_t [getConsoleHeight](#) ()
- int32\_t [getConsolePositionX](#) ()
- int32\_t [getConsolePositionY](#) ()
- void [setConsoleHidden](#) (bool hidden)  
*hide console window*
- bool [isConsoleHidden](#) ()
- void [main](#) (void \*instance, void \*prev\_instance, wchar\_t \*command, int32\_t show\_mode, [Main](#) \*main)

#### 4.23.1 Detailed Description

[Windows](#)

## 5 Class Documentation

### 5.1 Tellusim::App Class Reference

```
#include <TellusimApp.h>
```

#### Public Types

- enum {  
**Version\_19** = 20221010,  
**Version\_20** = 20221109,  
**Version\_21** = 20221122,  
**Version\_22** = 20221222,  
**Version\_23** = 20230117,  
**Version\_24** = 20230217,  
**Version\_25** = 20230402,  
**Version\_26** = 20230509,  
**Version\_27** = 20230612,  
**Version\_28** = 20230718,  
**Version\_29** = 20230824,  
**Version\_30** = 20231029,  
**Version\_31** = 20231113,  
**Version\_32** = 20231212,  
**Version\_33** = 20240116,  
**Version\_34** = 20240216,  
**Version\_35** = 20240320,  
**Version\_36** = 20240427,  
**Version\_37** = 20240515,  
**Version** = Version\_37 }

*Release version.*

#### Public Member Functions

- **App** (int32\_t argc, char \*\*argv)
- void **clear** ()  
*clear application*
- Platform **getPlatform** () const  
*command line parameters*
- uint32\_t **getDevice** () const
- uint32\_t **getWidth** () const
- uint32\_t **getHeight** () const
- uint32\_t **getMultisample** () const
- uint32\_t **getNumArguments** () const  
*command line arguments*
- const **String** & **getArgument** (uint32\_t num) const
- const Array< **String** > & **getArguments** () const
- bool **isArgument** (const char \*name) const
- const **String** & **getArgument** (const char \*name) const
- bool **create** (Platform platform=PlatformUnknown, uint32\_t version=Version)  
*create application*

## Static Public Member Functions

- static void [setPlatform](#) (Platform platform, uint32\_t device=Maxu32)  
*default parameters*
- static void **setSize** (uint32\_t width, uint32\_t height, uint32\_t multisample=0)
- static bool [isBuildDebug](#) ()  
*binary info*
- static bool **isBuildFloat64** ()
- static [String](#) **getBuildDate** ()
- static [String](#) **getBuildInfo** ()
- static uint32\_t **getVersion** ()
- static uint64\_t **getAPIHash** ()

## 5.1.1 Detailed Description

Application

## 5.2 Tellusim::Archive Class Reference

```
#include <format/TellusimArchive.h>
```

## Public Member Functions

- bool [open](#) (const char \*name, const char \*type=nullptr)  
*open/close archive*
- bool **open** (const [String](#) &name, const char \*type=nullptr)
- bool **open** ([Stream](#) &stream, const char \*type=nullptr)
- void **close** ()
- bool [isOpen](#) () const  
*archive status*
- [String](#) **getName** () const
- uint32\_t [getNumFiles](#) () const  
*files list*
- [String](#) **getFileName** (uint32\_t index) const
- uint64\_t **getFileMTime** (uint32\_t index) const
- size\_t **getFileSize** (uint32\_t index) const
- uint32\_t [findFile](#) (const char \*name) const  
*find file*
- uint32\_t **findFile** (const [String](#) &name) const
- bool **isFile** (const char \*name) const
- bool **isFile** (const [String](#) &name) const
- [Stream](#) [openFile](#) (const char \*name) const  
*open file*
- [Stream](#) **openFile** (const [String](#) &name) const
- [Stream](#) **openFile** (uint32\_t index) const

## 5.2.1 Detailed Description

[Archive](#)

### 5.3 Tellusim::ArchiveStream Class Reference

```
#include <format/TellusimArchive.h>
```

#### Public Member Functions

- virtual [ArchiveStream](#) \* [instance](#) () const =0  
*create instance*
- virtual void **destructor** ([ArchiveStream](#) \*[instance](#)) const =0
- virtual bool [open](#) ([Stream](#) &stream, const char \*name)=0  
*open archive*
- virtual uint32\_t [getNumFiles](#) () const =0  
*files list*
- virtual const [String](#) & **getFileName** (uint32\_t index) const =0
- virtual uint64\_t **getFileMTime** (uint32\_t index) const =0
- virtual size\_t **getFileSize** (uint32\_t index) const =0
- virtual [Stream](#) [openFile](#) (uint32\_t index)=0  
*open file*

#### Static Public Member Functions

- static bool [check](#) (const [String](#) &name)  
*archive stream formats*
- static [String](#) [getFormats](#) ()  
*list of supported formats*

#### Protected Member Functions

- **ArchiveStream** (const char \*name)
- **ArchiveStream** (const InitializerList< const char \*> &names)

#### 5.3.1 Detailed Description

[ArchiveStream](#) class

### 5.4 Tellusim::Async Class Reference

```
#include <core/TellusimAsync.h>
```

#### Classes

- class [Task](#)  
[Task](#).

## Public Member Functions

- bool **init** (uint32\_t num=0)  
*initialize threads*
- bool **shutdown** ()  
*shutdown threads*
- bool **isInitialized** () const  
*check status*
- uint32\_t **getNumThreads** () const
- void **append** (const Task &task)  
*temporary tasks*
- void **clear** ()
- uint32\_t **getNumTasks** (bool check=false) const  
*number of temporary tasks*
- Task **run** (uint32\_t mask)  
*run function*
- Task **run** (uint64\_t mask=~0ull)
- template<class Func >  
Task **run** (const Func &func, uint64\_t mask=~0ull)
- template<class Func , class A0 >  
Task **run** (const Func &func, A0 a0, uint64\_t mask=~0ull)
- template<class Func , class A0 , class A1 >  
Task **run** (const Func &func, A0 a0, A1 a1, uint64\_t mask=~0ull)
- template<class Func , class A0 , class A1 , class A2 >  
Task **run** (const Func &func, A0 a0, A1 a1, A2 a2, uint64\_t mask=~0ull)
- template<class Func , class A0 , class A1 , class A2 , class A3 >  
Task **run** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3, uint64\_t mask=~0ull)
- template<class Func , class A0 , class A1 , class A2 , class A3 , class A4 >  
Task **run** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3, A4 a4, uint64\_t mask=~0ull)
- template<class Func , class A0 , class A1 , class A2 , class A3 , class A4 , class A5 >  
Task **run** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3, A4 a4, A5 a5, uint64\_t mask=~0ull)
- template<class Func , class A0 , class A1 , class A2 , class A3 , class A4 , class A5 , class A6 >  
Task **run** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3, A4 a4, A5 a5, A6 a6, uint64\_t mask=~0ull)
- template<class Func , class A0 , class A1 , class A2 , class A3 , class A4 , class A5 , class A6 , class A7 >  
Task **run** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3, A4 a4, A5 a5, A6 a6, A7 a7, uint64\_t mask=~0ull)
- template<class Func >  
Task **run** (const Function< Func > &func, uint64\_t mask=~0ull)
- bool **check** (const Task \*tasks, uint32\_t num) const  
*check completion status*
- bool **check** (const Array< Task > &tasks) const
- bool **check** () const
- bool **wait** (const Task \*tasks, uint32\_t num) const  
*waiting for the completion*
- bool **wait** (const Array< Task > &tasks) const
- bool **wait** () const

## Static Public Member Functions

- static uint32\_t **getNumCores** ()  
*number of threads*

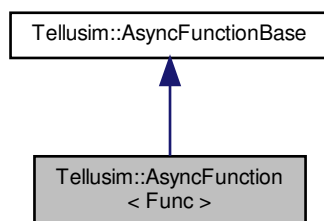
### 5.4.1 Detailed Description

[Async](#) task scheduler

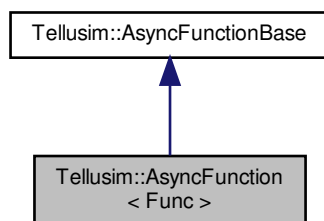
## 5.5 Tellusim::AsyncFunction< Func > Class Template Reference

```
#include <core/TellusimAsync.h>
```

Inheritance diagram for Tellusim::AsyncFunction< Func >:



Collaboration diagram for Tellusim::AsyncFunction< Func >:



### Public Types

- using [Ret](#) = typename Func::Ret  
*function return type*

### Public Member Functions

- **AsyncFunction** (const Func &func)
- virtual void [run](#) ()  
*run function*
- virtual void \* [get](#) ()  
*return value pointer*



## Additional Inherited Members

## 5.5.1 Detailed Description

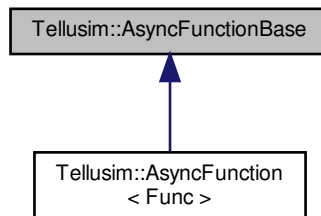
```
template<class Func>
class Tellusim::AsyncFunction< Func >
```

[Async](#) function template

## 5.6 Tellusim::AsyncFunctionBase Class Reference

```
#include <core/TellusimAsync.h>
```

Inheritance diagram for Tellusim::AsyncFunctionBase:



## Public Member Functions

- virtual void [run](#) ()=0  
*run function*
- virtual void \* [get](#) ()=0  
*function result*

## Static Public Member Functions

- static void [release](#) ([AsyncFunctionBase](#) \*func)  
*release function pointer*

## 5.6.1 Detailed Description

[Async](#) function base class

## 5.7 Tellusim::AsyncFunctionRet&lt; Type &gt; Struct Template Reference

```
#include <core/TellusimAsync.h>
```

## Public Types

- using **Ret** = Type

## 5.7.1 Detailed Description

```
template<class Type>
struct Tellusim::AsyncFunctionRet< Type >
```

[Async](#) function return type

## 5.8 Tellusim::AsyncFunctionRet&lt; void &gt; Struct Template Reference

## Public Types

- using **Ret** = void \*

## 5.9 Tellusim::Atlas&lt; Type &gt; Class Template Reference

```
#include <geometry/TellusimAtlas.h>
```

## Classes

- struct [Node](#)  
*Atlas Node.*

## Public Types

- enum { **Axes** = Vector::Size }
- using **Bound** = Type
- using **Vector** = typename Type::Vector

## Public Member Functions

- [Atlas](#) ()  
*constructors*
- **Atlas** (const [Atlas](#) &atlas)
- **Atlas** ([Atlas](#) &&atlas)
- **Atlas** (const Vector &size)
- **Atlas** (const Bound &bound)
- void [clear](#) ()  
*clear atlas*
- void [set](#) (const Vector &size)  
*set atlas size*
- void **set** (const Bound &bound)
- [Atlas](#) & **operator=** (const [Atlas](#) &atlas)  
*assignment operators*
- [Atlas](#) & **operator=** ([Atlas](#) &&atlas)
- const [Node](#) \* [getRoot](#) () const  
*atlas root*
- [Node](#) \* [insert](#) (const Vector &size)  
*insert node into the atlas*
- bool [remove](#) ([Node](#) \*node)  
*remove node from the atlas*

## 5.9.1 Detailed Description

```
template<class Type>
class Tellusim::Atlas< Type >
```

[Atlas](#) [utils](#)

## 5.10 Tellusim::AtomicArray&lt; Type &gt; Class Template Reference

```
#include <core/TellusimAtomic.h>
```

## Public Member Functions

- **AtomicArray** (uint32\_t size)
- **AtomicArray** (uint32\_t size, const Type \*array)
- **AtomicArray** (uint32\_t size, const Type &value)
- **AtomicArray** ([AtomicArray](#) &array)
- void [reserve](#) (uint32\_t size)  
*resize array*
- void **resize** (uint32\_t size, bool [reserve](#)=false, bool discard=false)
- void **resize** (uint32\_t size, const Type &value, bool [reserve](#)=false)
- void [release](#) ()  
*clear array*
- void **clear** ()
- void [swap](#) ([AtomicArray](#) &array)  
*swap arrays*
- void [copy](#) (const Type \*array, uint32\_t size)  
*copy value*
- void **copy** ([AtomicArray](#) &array)
- [AtomicArray](#) & **operator=** ([AtomicArray](#) &array)  
*assignment operator*
- void [append](#) (const Type &value)  
*append value*
- [AtomicArray](#) & **append** (const Type \*array, uint32\_t size)
- [AtomicArray](#) & **append** ([AtomicArray](#) &array)
- void [removeFast](#) (uint32\_t pos, uint32\_t len=1)  
*remove value*
- void **remove** (uint32\_t pos, uint32\_t len=1)
- bool [empty](#) ()  
*array info*
- **operator bool** ()
- uint32\_t **bytes** ()
- uint32\_t **memory** ()
- uint32\_t **size** ()
- Type \* [get](#) ()  
*array data*
- Type & **operator[]** (uint32\_t index)
- Type & **get** (uint32\_t index)
- template<class T >  
uint32\_t [findIndex](#) (const T &value)  
*array data*
- Type \* [begin](#) ()  
*array iterators*
- Type \* **end** ()

### 5.10.1 Detailed Description

```
template<class Type>
class Tellusim::AtomicArray< Type >
```

Atomic Array

## 5.11 Tellusim::Atomici32 Struct Reference

```
#include <core/TellusimAtomic.h>
```

### Public Member Functions

- **Atomici32** (int32\_t value=0)
- **operator int32\_t** ()
- **Atomici32 & operator=** (int32\_t value)
- int32\_t **operator++** ()
  - atomic operators*
- int32\_t **operator--** ()
- int32\_t **operator++** (int32\_t)
- int32\_t **operator--** (int32\_t)
- int32\_t **operator+=** (int32\_t v)
- int32\_t **operator-=** (int32\_t v)
- int32\_t **operator &=** (int32\_t v)
- int32\_t **operator|=** (int32\_t v)
- void **set** (int32\_t v)
  - atomic functions*
- int32\_t **get** ()
- bool **cas** (int32\_t old\_value, int32\_t new\_value)

### Public Attributes

- volatile int32\_t **value**

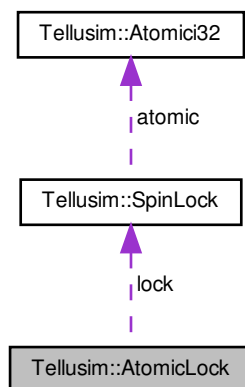
### 5.11.1 Detailed Description

32-bit integer Atomic

## 5.12 Tellusim::AtomicLock Struct Reference

```
#include <core/TellusimAtomic.h>
```

Collaboration diagram for Tellusim::AtomicLock:



### Public Member Functions

- **AtomicLock** ([SpinLock](#) &lock)

### Public Attributes

- [SpinLock](#) & **lock**

#### 5.12.1 Detailed Description

[AtomicLock](#) class

## 5.13 Tellusim::AtomicPtr< Type > Struct Template Reference

```
#include <core/TellusimAtomic.h>
```

## Public Member Functions

- **AtomicPtr** (Type \*ptr=nullptr)
- **AtomicPtr** & **operator=** (Type \*ptr)
- **operator bool** ()
- Type & **operator[]** (int32\_t index)
- Type & **operator[]** (uint32\_t index)
- Type \* **operator++** ()
- atomic operators*
- Type \* **operator--** ()
- Type \* **operator++** (int32\_t)
- Type \* **operator--** (int32\_t)
- Type \* **operator+=** (size\_t v)
- Type \* **operator-=** (size\_t v)
- void **set** (Type \*p)
- atomic functions*
- Type \* **get** ()
- bool **cas** (Type \*old\_ptr, Type \*new\_ptr)

## Public Attributes

- volatile Type \* **ptr**

## 5.13.1 Detailed Description

```
template<class Type>
struct Tellusim::AtomicPtr< Type >
```

Atomic pointer

## 5.14 Tellusim::BitonicSort Class Reference

```
#include <parallel/TellusimBitonicSort.h>
```

## Classes

- struct **DispatchParameters**

## Public Types

- enum **Mode** {  
    **ModeSingle** = 0,  
    **ModeMultiple**,  
    **NumModes** }
- Sort modes.*
- enum **Flags** {  
    **FlagNone** = 0,  
    **FlagSingle** = (1 << ModeSingle),  
    **FlagMultiple** = (1 << ModeMultiple),  
    **FlagIndirect** = (1 << (NumModes + 0)),  
    **FlagOrder** = (1 << (NumModes + 1)),  
    **FlagsAll** = (FlagSingle | FlagMultiple | FlagIndirect | FlagOrder) }
- Sort flags.*

## Public Member Functions

- void **clear** ()  
*clear sort*
- bool **isCreated** (**Flags** flags) const  
*check sort*
- uint32\_t **getDataSize** () const  
*sort parameters*
- uint32\_t **getGroupSize** () const
- uint32\_t **getSortElements** () const
- uint32\_t **getMaxRegions** () const
- bool **create** (const **Device** &device, **Mode** mode, uint32\_t size, uint32\_t groups=256, uint32\_t regions=1, **Async** \*async=nullptr)
- bool **create** (const **Device** &device, **Flags** flags, uint32\_t size, uint32\_t groups=256, uint32\_t regions=1, **Async** \*async=nullptr)
- bool **dispatch** (**Compute** &compute, **Buffer** &data, uint32\_t keys\_offset, uint32\_t data\_offset, uint32\_t size, **Flags** flags=FlagNone)
- bool **dispatch** (**Compute** &compute, **Buffer** &data, uint32\_t count, const uint32\_t \*keys\_offsets, const uint32\_t \*data\_offsets, const uint32\_t \*sizes, **Flags** flags=FlagNone)
- bool **dispatchIndirect** (**Compute** &compute, **Buffer** &data, **Buffer** &dispatch, uint32\_t offset, **Flags** flags=FlagNone)
- bool **dispatchIndirect** (**Compute** &compute, **Buffer** &data, uint32\_t count, **Buffer** &dispatch, uint32\_t offset, **Flags** flags=FlagNone)
- bool **dispatchIndirect** (**Compute** &compute, **Buffer** &data, **Buffer** &count, **Buffer** &dispatch, uint32\_t count\_offset, uint32\_t dispatch\_offset, **Flags** flags=FlagNone)

## 5.14.1 Detailed Description

**BitonicSort** class

## 5.14.2 Member Function Documentation

## 5.14.2.1 create()

```
bool Tellusim::BitonicSort::create (
    const Device & device,
    Mode mode,
    uint32_t size,
    uint32_t groups = 256,
    uint32_t regions = 1,
    Async * async = nullptr )
```

create bitonic sort

## Parameters

<i>size</i>	Maximum number of sorted elements.
<i>groups</i>	Bitonic sort group size.
<i>regions</i>	Maximum number of multiple regions.

### 5.14.2.2 `dispatch()` [1/2]

```
bool Tellusim::BitonicSort::dispatch (
    Compute & compute,
    Buffer & data,
    uint32_t keys_offset,
    uint32_t data_offset,
    uint32_t size,
    Flags flags = FlagNone )
```

dispatch inplace single bitonic sort

#### Parameters

<i>data</i>	Buffer of uint32_t data elements to sort.
<i>keys_offset</i>	Keys elements offset index (2 aligned).
<i>data_offset</i>	Data elements offset index (2 aligned).
<i>size</i>	Number of uint32_t elements to sort.

### 5.14.2.3 `dispatch()` [2/2]

```
bool Tellusim::BitonicSort::dispatch (
    Compute & compute,
    Buffer & data,
    uint32_t count,
    const uint32_t * keys_offsets,
    const uint32_t * data_offsets,
    const uint32_t * sizes,
    Flags flags = FlagNone )
```

dispatch inplace multiple bitonic sorts

#### Parameters

<i>data</i>	Buffer of uint32_t data elements to sort.
<i>count</i>	Number of regions to sort.
<i>keys_offsets</i>	Keys elements offset index (2 aligned).
<i>data_offsets</i>	Data elements offset index (2 aligned).
<i>sizes</i>	Number of uint32_t elements to sort.

### 5.14.2.4 `dispatchIndirect()` [1/3]

```
bool Tellusim::BitonicSort::dispatchIndirect (
    Compute & compute,
    Buffer & data,
    Buffer & dispatch,
```



```
uint32_t offset,
Flags flags = FlagNone )
```

dispatch inplace single indirect local bitonic sort

#### Parameters

<i>data</i>	Buffer of uint32_t data elements to sort.
<i>dispatch</i>	Dispatch indirect buffer.
<i>offset</i>	Dispatch indirect buffer offset.

#### 5.14.2.5 dispatchIndirect() [2/3]

```
bool Tellusim::BitonicSort::dispatchIndirect (
    Compute & compute,
    Buffer & data,
    uint32_t count,
    Buffer & dispatch,
    uint32_t offset,
    Flags flags = FlagNone )
```

dispatch inplace multiple indirect local bitonic sorts

#### Parameters

<i>data</i>	Buffer of uint32_t data elements to sort.
<i>count</i>	Number of regions to sort.
<i>dispatch</i>	Dispatch indirect buffer.
<i>offset</i>	Dispatch indirect buffer offset.

#### 5.14.2.6 dispatchIndirect() [3/3]

```
bool Tellusim::BitonicSort::dispatchIndirect (
    Compute & compute,
    Buffer & data,
    Buffer & count,
    Buffer & dispatch,
    uint32_t count_offset,
    uint32_t dispatch_offset,
    Flags flags = FlagNone )
```

dispatch inplace multiple indirect local bitonic sorts

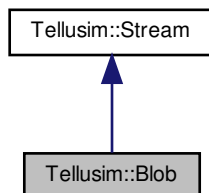
#### Parameters

<i>data</i>	Buffer of uint32_t data elements to sort.
<i>count</i>	Count indirect buffer.
<i>dispatch</i>	Dispatch indirect buffer.
<i>count_offset</i>	Count indirect buffer offset.
<i>dispatch_offset</i>	Dispatch indirect buffer offset.

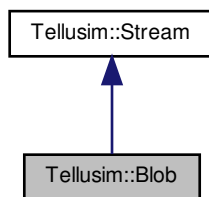
### 5.15 Tellusim::Blob Class Reference

```
#include <core/TellusimBlob.h>
```

Inheritance diagram for Tellusim::Blob:



Collaboration diagram for Tellusim::Blob:



#### Public Member Functions

- **Blob** (const char \*name=nullptr)
- **Blob** (const [String](#) &name)
- **Blob** (const uint8\_t \*data, size\_t size, const char \*name=nullptr)
- **Blob** (const uint8\_t(\*blob)[256], const char \*name=nullptr)
- **Blob** (const [Blob](#) &blob, bool [move](#))
- void [release](#) ()
  - clear blob*
- void **clear** ()
- void [setName](#) (const char \*name)
  - blob name*
- void **setName** (const [String](#) &name)
- void [setSize](#) (size\_t size)
  - blob data*
- void **setCapacity** (size\_t size)
- size\_t **getCapacity** () const

- bool **setData** (const uint8\_t \*data, size\_t size)
- bool **setData** (const uint8\_t(\*blob)[256])
- bool **setData** (const [Blob](#) &blob)
- const uint8\_t \* **getData** () const
- uint8\_t \* **getData** ()
- [String](#) **encodeBase64** (size\_t size=0)  
*base64 encoding*
- bool **decodeBase64** (const char \*src)
- void **getMD5** (uint32\_t hash[4], size\_t size=0)  
*message digest algorithm*
- [String](#) **getMD5** (size\_t size=0)
- void **getSHA1** (uint32\_t hash[5], size\_t size=0)  
*secure hash algorithm*
- [String](#) **getSHA1** (size\_t size=0)

#### Static Public Member Functions

- static [String](#) **getMD5** (const [String](#) &str)
- static [String](#) **getMD5** (const void \*src, size\_t size)
- static [String](#) **getMD5** ([Stream](#) &src, size\_t size=0)
- static [String](#) **getSHA1** (const [String](#) &str)
- static [String](#) **getSHA1** (const void \*src, size\_t size)
- static [String](#) **getSHA1** ([Stream](#) &src, size\_t size=0)

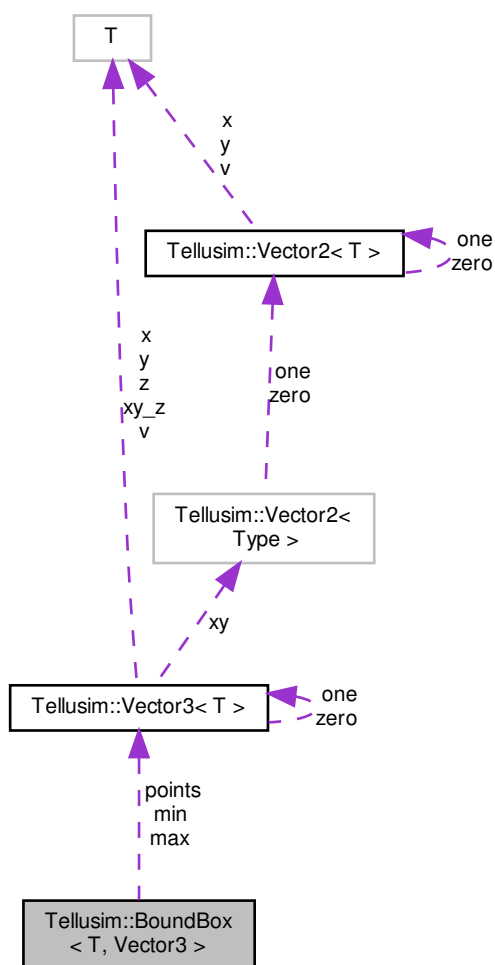
#### 5.15.1 Detailed Description

[Blob](#) class

## 5.16 Tellusim::BoundingBox< T, Vector3 > Struct Template Reference

```
#include <geometry/TellusimBounds.h>
```

Collaboration diagram for Tellusim::BoundingBox< T, Vector3 >:



#### Public Types

- using **Vector** = `Vector3`
- using **Vector2** = `Tellusim::Vector2< Type >`
- using **BoundSphere** = `Tellusim::BoundSphere< Type, Vector3 >`

#### Public Member Functions

- **BoundingBox** (const `BoundingBox` &bb)
- **BoundingBox** (const `Vector3` &bb\_min, const `Vector3` &bb\_max)
- **BoundingBox** (const `Vector3` &bs\_center, Type bs\_radius)
- **BoundingBox** (const `BoundingBox` &bb, const `BoundSphere` &bs)
- **BoundingBox** (const `BoundSphere` &bs)
- template<class CType , class CVector >  
**BoundingBox** (const `Tellusim::BoundingBox< CType, CVector >` &bb)

- `template<class CType , class CVector >`  
**BoundingBox** (const [Tellusim::BoundSphere](#)< CType, CVector > &bs)
- **BoundingBox** (const [Vector3](#) \*1 points, uint32\_t num\_points)
- void **clear** ()  
*clear bound box*
- bool **isValid** () const  
*check bound box*
- **operator bool** () const
- void **set** (const [Vector3](#) &bb\_min, const [Vector3](#) &bb\_max)  
*set bound box*
- void **set** (const [BoundingBox](#) &bb)
- void **set** (const [Vector3](#) &bs\_center, Type bs\_radius)  
*set bound sphere*
- void **set** (const [BoundSphere](#) &bs)
- void **set** (const [Vector3](#) &bb\_min, const [Vector3](#) &bb\_max, const [Vector3](#) &bs\_center, Type bs\_radius)  
*set minimal bound box*
- void **set** (const [BoundingBox](#) &bb, const [BoundSphere](#) &bs)
- void **set** (const [Vector3](#) \*1 points, uint32\_t num\_points)  
*set bound box*
- void **expand** (const [Vector3](#) &point)  
*expand by point*
- void **expand** (const [Vector3](#) &bb\_min, const [Vector3](#) &bb\_max)  
*expand by bound box*
- void **expand** (const [BoundingBox](#) &bb)
- void **expand** (const [Vector3](#) &bs\_center, Type bs\_radius)  
*expand by bound sphere*
- void **expand** (const [BoundSphere](#) &bs)
- void **expand** (const [Vector3](#) &bb\_min, const [Vector3](#) &bb\_max, const [Vector3](#) &bs\_center, Type bs\_radius)  
*expand by minimal bounds*
- void **expand** (const [BoundingBox](#) &bb, const [BoundSphere](#) &bs)
- void **shrink** (const [Vector3](#) &bb\_min, const [Vector3](#) &bb\_max)  
*shrink by bound box*
- void **shrink** (const [BoundingBox](#) &bb)
- void **shrink** (const [Vector3](#) &bs\_center, Type bs\_radius)  
*shrink by bound sphere*
- void **shrink** (const [BoundSphere](#) &bs)
- bool **inside** (const [Vector3](#) &point) const  
*inside point*
- bool **inside** (const [Vector3](#) &bb\_min, const [Vector3](#) &bb\_max) const  
*inside bound box*
- bool **inside** (const [BoundingBox](#) &bb) const
- bool **inside** (const [Vector3](#) &bs\_center, Type bs\_radius) const  
*inside bound sphere*
- bool **inside** (const [BoundSphere](#) &bs) const
- Type **distance** (const [Vector3](#) &point) const  
*signed distance to the bound box*
- [Vector2](#) **trace** (const [Vector3](#) &point, const [Vector3](#) &idirection) const  
*bound box ray tracing*
- [Vector3](#) **getCenter** () const  
*to bound sphere*
- Type **getRadius** () const
- [Vector3](#) **getSize** () const

*bound box parameters*

- const [Vector3](#) & **getMin** () const
- const [Vector3](#) & **getMax** () const
- const [Vector3](#) \* **getPoints** () const
- Type **getVolume** (Type threshold=1e-8f) const

#### Public Attributes

```

•
union {
    struct {
        Vector3 min
        Vector3 max
    }
    Vector3 points [2]
};

```

#### 5.16.1 Detailed Description

```

template<class T, class Vector3 = Tellusim::Vector3<T>>
struct Tellusim::BoundingBox< T, Vector3 >

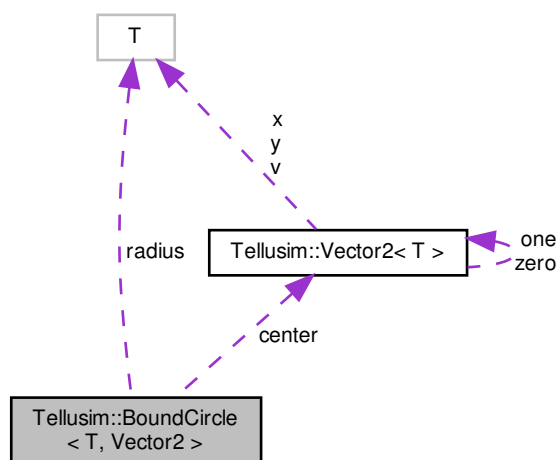
```

[BoundingBox](#) class

#### 5.17 Tellusim::BoundCircle< T, Vector2 > Struct Template Reference

```
#include <geometry/TellusimBounds.h>
```

Collaboration diagram for Tellusim::BoundCircle< T, Vector2 >:



## Public Types

- using **Vector** = [Vector2](#)
- using **BoundRect** = [Tellusim::BoundRect](#)< Type, [Vector2](#) >

## Public Member Functions

- **BoundCircle** (const [BoundCircle](#) &bc)
- **BoundCircle** (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max)
- **BoundCircle** (const [Vector2](#) &bc\_center, Type bc\_radius)
- **BoundCircle** (const [BoundRect](#) &br)
- template<class CType, class CVector >  
**BoundCircle** (const [Tellusim::BoundRect](#)< CType, CVector > &br)
- template<class CType, class CVector >  
**BoundCircle** (const [Tellusim::BoundCircle](#)< CType, CVector > &bc)
- **BoundCircle** (const [Vector2](#) \*1 points, uint32\_t num\_points)
- void [clear](#) ()  
*clear bound circle*
- bool [isValid](#) () const  
*check bound circle*
- **operator bool** () const
- void [set](#) (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max)  
*set bound rect*
- void [set](#) (const [BoundRect](#) &br)
- void [set](#) (const [Vector2](#) &bc\_center, Type bc\_radius)  
*set bound circle*
- void [set](#) (const [BoundCircle](#) &bc)
- void [set](#) (const [Vector2](#) \*1 points, uint32\_t num\_points)  
*set bound circle*
- void [expand](#) (const [Vector2](#) &point)  
*expand by point*
- void [expand](#) (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max)  
*expand by bound rect*
- void [expand](#) (const [BoundRect](#) &br)
- void [expand](#) (const [Vector2](#) &bc\_center, Type bc\_radius)  
*expand by bound circle*
- void [expand](#) (const [BoundCircle](#) &bc)
- void [expand](#) (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max, const [Vector2](#) &bc\_center, Type bc\_radius)  
*expand by minimal bounds*
- void [expand](#) (const [BoundRect](#) &br, const [BoundCircle](#) &bc)
- void [expandRadius](#) (const [Vector2](#) &point)  
*expand radius by point*
- void [expandRadius](#) (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max)  
*expand radius by bound rect*
- void [expandRadius](#) (const [BoundRect](#) &br)
- void [expandRadius](#) (const [Vector2](#) &bc\_center, Type bc\_radius)  
*expand radius by bound circle*
- void [expandRadius](#) (const [BoundCircle](#) &bc)
- bool [inside](#) (const [Vector2](#) &point) const  
*inside point*
- bool [inside](#) (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max) const  
*inside bound rect*

- bool **inside** (const [BoundRect](#) &br) const
- bool **inside** (const [Vector2](#) &bc\_center, Type bc\_radius) const  
*inside bound circle*
- bool **inside** (const [BoundCircle](#) &bc) const
- Type **distance** (const [Vector2](#) &point) const  
*signed distance to bound circle*
- [Vector2](#) **getMin** () const  
*to bound rect*
- [Vector2](#) **getMax** () const
- const [Vector2](#) & **getCenter** () const  
*bound circle parameters*
- Type **getRadius** () const
- Type **getArea** () const

#### Public Attributes

- [Vector2](#) **center**
- Type **radius**

#### 5.17.1 Detailed Description

```
template<class T, class Vector2 = Tellusim::Vector2<T>>
struct Tellusim::BoundCircle< T, Vector2 >
```

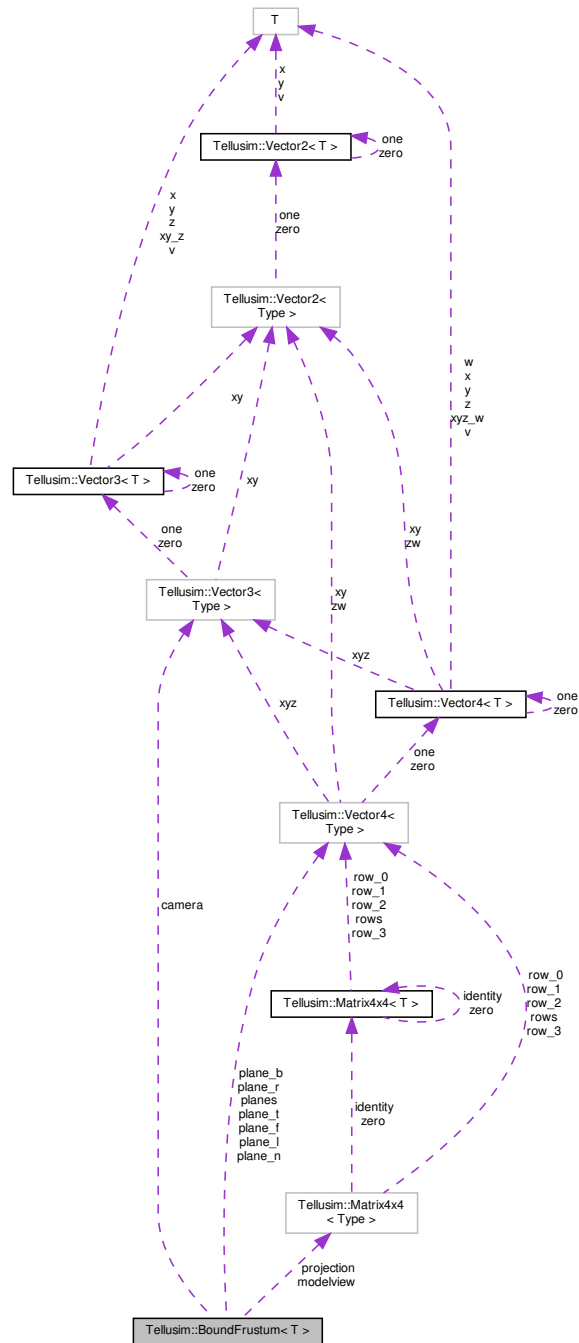
[BoundCircle](#) class

### 5.18 Tellusim::BoundFrustum< T > Struct Template Reference

```
#include <geometry/TellusimBounds.h>
```



Collaboration diagram for Tellusim::BoundFrustum< T >:



## Public Types

- using **Vector3** = `Tellusim::Vector3< Type >`
- using **Vector4** = `Tellusim::Vector4< Type >`
- using **Matrix4x4** = `Tellusim::Matrix4x4< Type >`

## Public Member Functions

- **BoundFrustum** (const [Matrix4x4](#) &projection, const [Matrix4x4](#) &modelview, Type aspect=1.0f)
- **BoundFrustum** (const [BoundFrustum](#) &bf)
- template<class CType >  
**BoundFrustum** (const [BoundFrustum](#)< CType > &bf)
- void **set** (const [Matrix4x4](#) &p, const [Matrix4x4](#) &m, Type aspect=1.0f)  
*bound frustum from matrix*
- template<class BType , class BVector >  
bool **inside4** (const [BoundingBox](#)< BType, BVector > &bb) const  
*inside bound box*
- template<class BType , class BVector >  
bool **inside** (const [BoundingBox](#)< BType, BVector > &bb) const
- template<class BType , class BVector >  
bool **insideAll4** (const [BoundingBox](#)< BType, BVector > &bb) const
- template<class BType , class BVector >  
bool **insideAll** (const [BoundingBox](#)< BType, BVector > &bb) const
- template<class BType , class BVector >  
bool **inside4** (const [BoundSphere](#)< BType, BVector > &bs) const  
*inside bound sphere*
- template<class BType , class BVector >  
bool **inside** (const [BoundSphere](#)< BType, BVector > &bs) const
- template<class BType , class BVector >  
bool **insideAll4** (const [BoundSphere](#)< BType, BVector > &bs) const
- template<class BType , class BVector >  
bool **insideAll** (const [BoundSphere](#)< BType, BVector > &bs) const
- const [Matrix4x4](#) &**getProjection** () const  
*bound frustum parameters*
- const [Matrix4x4](#) &**getModelview** () const
- const [Vector3](#) &**getCamera** () const

## Public Attributes

- [Matrix4x4](#) **projection**
- [Matrix4x4](#) **modelview**
- [Vector3](#) **camera**
- 

```
union {
    struct {
        Vector4 plane_l
        Vector4 plane_r
        Vector4 plane_b
        Vector4 plane_t
        Vector4 plane_n
        Vector4 plane_f
    }
    Vector4 planes [6]
};
```

-

```

union {
    struct {
        uint8_t sign_l [4]
        uint8_t sign_r [4]
        uint8_t sign_b [4]
        uint8_t sign_t [4]
        uint8_t sign_n [4]
        uint8_t sign_f [4]
    }
    uint8_t signs [6][4]
};

```

### 5.18.1 Detailed Description

```

template<class T>
struct Tellusim::BoundFrustum< T >

```

[BoundFrustum](#) class

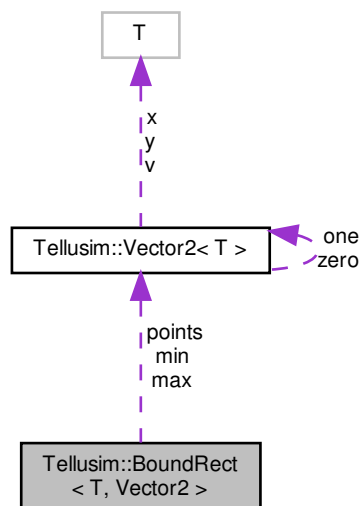
## 5.19 Tellusim::BoundRect< T, Vector2 > Struct Template Reference

```

#include <geometry/TellusimBounds.h>

```

Collaboration diagram for Tellusim::BoundRect< T, Vector2 >:



### Public Types

- using **Vector** = [Vector2](#)
- using **BoundCircle** = [Tellusim::BoundCircle](#)< Type, [Vector2](#) >

## Public Member Functions

- **BoundRect** (const [BoundRect](#) &br)
- **BoundRect** (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max)
- **BoundRect** (const [Vector2](#) &bc\_center, Type bc\_radius)
- **BoundRect** (const [BoundRect](#) &br, const [BoundCircle](#) &bc)
- **BoundRect** (const [BoundCircle](#) &bc)
- template<class CType, class CVector >  
**BoundRect** (const [Tellusim::BoundRect](#)< CType, CVector > &br)
- template<class CType, class CVector >  
**BoundRect** (const [Tellusim::BoundCircle](#)< CType, CVector > &bc)
- **BoundRect** (const [Vector2](#) \*1 points, uint32\_t num\_points)
- void [clear](#) ()  
*clear bound rect*
- bool [isValid](#) () const  
*check bound rect*
- **operator bool** () const
- void [set](#) (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max)  
*set bound rect*
- void [set](#) (const [BoundRect](#) &br)
- void [set](#) (const [Vector2](#) &bc\_center, Type bc\_radius)  
*set bound circle*
- void [set](#) (const [BoundCircle](#) &bc)
- void [set](#) (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max, const [Vector2](#) &bc\_center, const Type &bc\_radius)  
*set minimal bound rect*
- void [set](#) (const [BoundRect](#) &br, const [BoundCircle](#) &bc)
- void [set](#) (const [Vector2](#) \*1 points, uint32\_t num\_points)  
*set bound rect*
- void [expand](#) (const [Vector2](#) &point)  
*expand by point*
- void [expand](#) (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max)  
*expand by bound rect*
- void [expand](#) (const [BoundRect](#) &br)
- void [expand](#) (const [Vector2](#) &bc\_center, Type bc\_radius)  
*expand by bound circle*
- void [expand](#) (const [BoundCircle](#) &bc)
- bool [inside](#) (const [Vector2](#) &point) const  
*inside point*
- bool [inside](#) (const [Vector2](#) &br\_min, const [Vector2](#) &br\_max) const  
*inside bound rect*
- bool [inside](#) (const [BoundRect](#) &br) const
- bool [inside](#) (const [Vector2](#) &bc\_center, Type bc\_radius) const  
*inside bound circle*
- bool [inside](#) (const [BoundCircle](#) &bc) const
- Type [distance](#) (const [Vector2](#) &point) const  
*signed distance to the bound rect*
- [Vector2](#) [trace](#) (const [Vector2](#) &point, const [Vector2](#) &idirection) const  
*bound rect ray tracing*
- [Vector2](#) [getCenter](#) () const  
*to bound circle*
- Type [getRadius](#) () const
- [Vector2](#) [getSize](#) () const

*bound rect parameters*

- const [Vector2](#) & **getMin** () const
- const [Vector2](#) & **getMax** () const
- const [Vector2](#) \* **getPoints** () const
- Type **getArea** (Type threshold=1e-8f) const

#### Public Attributes

- ```

union {
    struct {
        Vector2 min
        Vector2 max
    }
    Vector2 points [2]
};

```

#### 5.19.1 Detailed Description

```

template<class T, class Vector2 = Tellusim::Vector2<T>>
struct Tellusim::BoundRect< T, Vector2 >

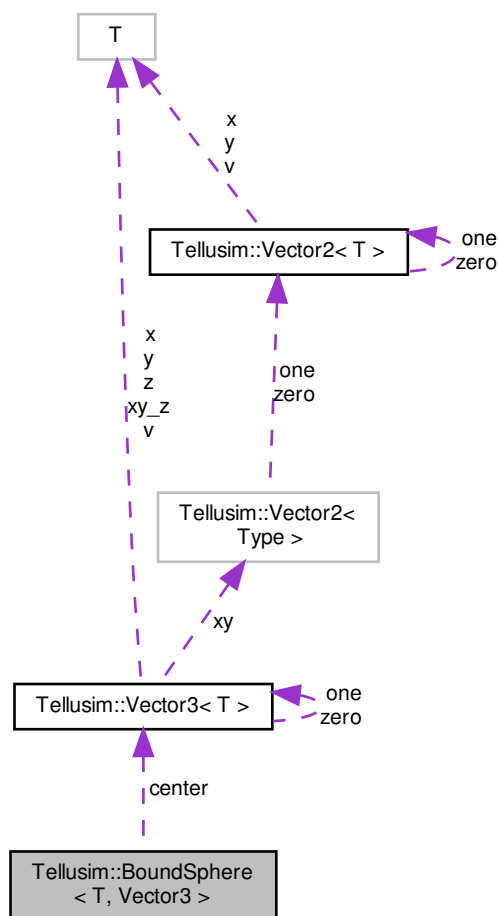
```

[BoundRect](#) class

## 5.20 Tellusim::BoundSphere< T, Vector3 > Struct Template Reference

```
#include <geometry/TellusimBounds.h>
```

Collaboration diagram for `Tellusim::BoundSphere< T, Vector3 >`:



#### Public Types

- using **Vector** = `Vector3`
- using **BoundingBox** = `Tellusim::BoundingBox< Type, Vector3 >`

#### Public Member Functions

- **BoundSphere** (const `BoundSphere` &bs)
- **BoundSphere** (const `Vector3` &bb\_min, const `Vector3` &bb\_max)
- **BoundSphere** (const `Vector3` &bs\_center, Type bs\_radius)
- **BoundSphere** (const `BoundingBox` &bb)
- template<class CType, class CVector >  
**BoundSphere** (const `Tellusim::BoundingBox< CType, CVector >` &bb)
- template<class CType, class CVector >  
**BoundSphere** (const `Tellusim::BoundSphere< CType, CVector >` &bs)
- **BoundSphere** (const `Vector3` \*1 points, uint32\_t num\_points)

- void **clear** ()  
*clear bound sphere*
- bool **isValid** () const  
*check bound sphere*
- **operator bool** () const
- void **set** (const **Vector3** &bb\_min, const **Vector3** &bb\_max)  
*set bound box*
- void **set** (const **BoundingBox** &bb)
- void **set** (const **Vector3** &bs\_center, Type bs\_radius)  
*set bound sphere*
- void **set** (const **BoundSphere** &bs)
- void **set** (const **Vector3** \*1 points, uint32\_t num\_points)  
*set bound sphere*
- void **expand** (const **Vector3** &point)  
*expand by point*
- void **expand** (const **Vector3** &bb\_min, const **Vector3** &bb\_max)  
*expand by bound box*
- void **expand** (const **BoundingBox** &bb)
- void **expand** (const **Vector3** &bs\_center, Type bs\_radius)  
*expand by bound sphere*
- void **expand** (const **BoundSphere** &bs)
- void **expand** (const **Vector3** &bb\_min, const **Vector3** &bb\_max, const **Vector3** &bs\_center, Type bs\_radius)  
*expand by minimal bounds*
- void **expand** (const **BoundingBox** &bb, const **BoundSphere** &bs)
- void **expandRadius** (const **Vector3** &point)  
*expand radius by point*
- void **expandRadius** (const **Vector3** &bb\_min, const **Vector3** &bb\_max)  
*expand radius by bound box*
- void **expandRadius** (const **BoundingBox** &bb)
- void **expandRadius** (const **Vector3** &bs\_center, Type bs\_radius)  
*expand radius by bound sphere*
- void **expandRadius** (const **BoundSphere** &bs)
- bool **inside** (const **Vector3** &point) const  
*inside point*
- bool **inside** (const **Vector3** &bb\_min, const **Vector3** &bb\_max) const  
*inside bound box*
- bool **inside** (const **BoundingBox** &bb) const
- bool **inside** (const **Vector3** &bs\_center, Type bs\_radius) const  
*inside bound sphere*
- bool **inside** (const **BoundSphere** &bs) const
- Type **distance** (const **Vector3** &point) const  
*signed distance to bound sphere*
- **Vector3** **getMin** () const  
*to bound box*
- **Vector3** **getMax** () const
- const **Vector3** & **getCenter** () const  
*bound sphere parameters*
- Type **getRadius** () const
- Type **getVolume** () const

## Public Attributes

- [Vector3](#) **center**
- Type **radius**

### 5.20.1 Detailed Description

```
template<class T, class Vector3 = Tellusim::Vector3<T>>
struct Tellusim::BoundSphere< T, Vector3 >
```

[BoundSphere](#) class

### 5.21 Tellusim::BrepModel Class Reference

```
#include <graphics/TellusimBrepModel.h>
```

## Classes

- struct [FaceParameters](#)

## Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagVerbose** = (1 << 0),  
**FlagCurve2** = (1 << 1),  
**FlagSurface2** = (1 << 2),  
**FlagMaterials** = (1 << 3),  
**FlagBufferTexture** = (1 << 4),  
**FlagBufferStorage** = (1 << 5),  
**FlagBufferTracing** = (1 << 6),  
**FlagBufferAddress** = (1 << 7),  
**FlagBufferTexel** = (1 << 8),  
**DefaultFlags** = (FlagVerbose | FlagMaterials | FlagBufferStorage),  
**NumFlags** = 9 }

*Model flags.*

- using [Face](#) = BrepFace::Type

*Brep face.*



## Public Member Functions

- void **clear** ()  
*clear model*
- bool **isCreated** () const  
*check model*
- **Flags** **getFlags** () const  
*model flags*
- bool **hasFlag** (**Flags** flags) const
- bool **hasFlags** (**Flags** flags) const
- bool **load** (const **Device** &device, const char \*name, **Flags** flags=DefaultFlags, **Async** \*async=nullptr)  
*load model*
- bool **load** (const **Device** &device, **Stream** &stream, **Flags** flags=DefaultFlags, **Async** \*async=nullptr)
- bool **create** (const **Device** &device, const Brep &brep, **Flags** flags=DefaultFlags)  
*create model*
- bool **create** (const **Device** &device, const BrepGeometry &geometry, **Flags** flags=DefaultFlags)
- bool **create** (const **Device** &device, const Array< BrepGeometry > &geometries, **Flags** flags=DefaultFlags)
- void **setBuffers** (**Command** &command, uint32\_t index=0, const **Pipeline** \*pipeline=nullptr) const  
*set model buffers*
- uint32\_t **getNumVertices** () const  
*vertices buffer*
- **Buffer** **getVertexBuffer** () const
- **Texture** **getVertexTexture** () const
- uint32\_t **getNumIndices** () const  
*indices buffer*
- **Buffer** **getIndexBuffer** () const
- **Texture** **getIndexTexture** () const
- uint32\_t **getNumRanges** () const  
*ranges buffer*
- **Buffer** **getRangeBuffer** () const
- **Texture** **getRangeTexture** () const
- uint32\_t **getNumPrimitives** () const  
*faces buffer*
- **Buffer** **getFaceBuffer** () const
- **Texture** **getFaceTexture** () const
- **Face** **getPrimitiveFace** (uint32\_t index) const  
*primitives parameters*
- uint32\_t **getPrimitiveIndex** (uint32\_t index) const
- uint32\_t **getPrimitiveGeometry** (uint32\_t index) const
- uint32\_t **getPrimitiveMaterial** (uint32\_t index) const
- uint32\_t **getBaseFaceMask** () const  
*primitive masks*
- uint32\_t **getWrapFaceMask** () const
- bool **hasBaseFace** (**Face** face) const
- bool **hasWrapFace** (**Face** face) const
- bool **hasFace** (**Face** face, bool wrap) const
- uint32\_t **getNumGeometries** () const  
*geometries*
- uint32\_t **getNumGeometryBaseIndices** (uint32\_t geometry, **Face** face) const
- uint32\_t **getNumGeometryWrapIndices** (uint32\_t geometry, **Face** face) const
- uint32\_t **getNumGeometryIndices** (uint32\_t geometry, **Face** face, bool wrap) const
- uint32\_t **getGeometryBaseIndex** (uint32\_t geometry, **Face** face) const
- uint32\_t **getGeometryWrapIndex** (uint32\_t geometry, **Face** face) const

- uint32\_t **getGeometryIndex** (uint32\_t geometry, [Face](#) face, bool wrap) const
- uint32\_t **getNumMaterials** (uint32\_t geometry) const  
*geometry materials*
- uint32\_t **getNumMaterialBaseIndices** (uint32\_t geometry, uint32\_t material, [Face](#) face) const
- uint32\_t **getNumMaterialWrapIndices** (uint32\_t geometry, uint32\_t material, [Face](#) face) const
- uint32\_t **getNumMaterialIndices** (uint32\_t geometry, uint32\_t material, [Face](#) face, bool wrap) const
- uint32\_t **getMaterialBaseIndex** (uint32\_t geometry, uint32\_t material, [Face](#) face) const
- uint32\_t **getMaterialWrapIndex** (uint32\_t geometry, uint32\_t material, [Face](#) face) const
- uint32\_t **getMaterialIndex** (uint32\_t geometry, uint32\_t material, [Face](#) face, bool wrap) const
- size\_t **getMemory** () const  
*memory usage*

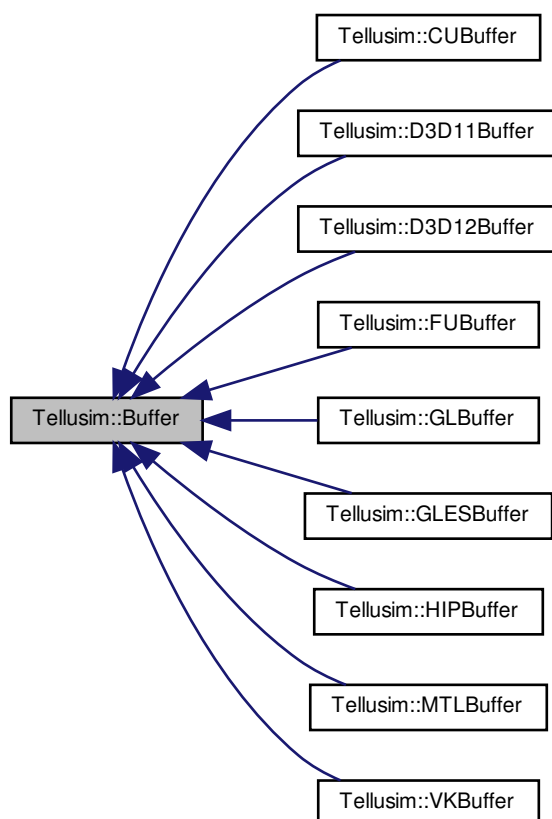
### 5.21.1 Detailed Description

[BrepModel](#) class

## 5.22 Tellusim::Buffer Class Reference

```
#include <platform/TellusimBuffer.h>
```

Inheritance diagram for Tellusim::Buffer:



## Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagRead** = (1 << 0),  
**FlagWrite** = (1 << 1),  
**FlagSource** = (1 << 2),  
**FlagSparse** = (1 << 3),  
**FlagShared** = (1 << 4),  
**FlagMapped** = (1 << 5),  
**FlagExtern** = (1 << 6),  
**FlagInterop** = (1 << 7),  
**FlagDynamic** = (1 << 8),  
**FlagUniform** = (1 << 9),  
**FlagStorage** = (1 << 10),  
**FlagAddress** = (1 << 11),  
**FlagTracing** = (1 << 12),  
**FlagScratch** = (1 << 13),  
**FlagBinding** = (1 << 14),  
**FlagIndirect** = (1 << 15),  
**FlagConditional** = (1 << 16),  
**FlagVertex** = (1 << 17),  
**FlagIndex** = (1 << 18),  
**FlagTexel** = (1 << 19),  
**FlagAccel** = (1 << 20),  
**DefaultFlags** = FlagNone,  
**NumFlags** = 21 }

*Buffer flags.*

## Public Member Functions

- Platform [getPlatform](#) () const  
*buffer platform*
- const char \* [getPlatformName](#) () const
- uint32\_t [getIndex](#) () const  
*buffer device index*
- void [clear](#) ()  
*clear buffer*
- bool [isCreated](#) () const  
*check buffer*
- void [setName](#) (const char \*name)  
*buffer name*
- [String](#) [getName](#) () const
- bool [create](#) ([Flags](#) flags, size\_t size, Format format=FormatUnknown)  
*create buffer*
- bool [isMapped](#) () const
- [Flags](#) [getFlags](#) () const  
*buffer flags*
- bool [hasFlag](#) ([Flags](#) flags) const
- bool [hasFlags](#) ([Flags](#) flags) const
- [String](#) [getFlagsName](#) () const
- Format [getFormat](#) () const  
*buffer format*
- const char \* [getFormatName](#) () const

- uint32\_t **getComponents** () const
- uint32\_t **getPixelSize** () const
- size\_t **getSize** ()  
*buffer size*
- size\_t **getPageSize** ()  
*sparse buffer page size*
- String **getDescription** () const  
*buffer description*

#### 5.22.1 Detailed Description

[Buffer](#) class

### 5.23 Tellusim::BufferTable Class Reference

```
#include <platform/TellusimBuffer.h>
```

#### Public Member Functions

- Platform **getPlatform** () const  
*table platform*
- const char \* **getPlatformName** () const
- uint32\_t **getIndex** () const  
*table device index*
- void **clear** ()  
*clear table*
- bool **isCreated** () const  
*check table*
- void **setName** (const char \*name)  
*table name*
- String **getName** () const
- bool **create** (uint32\_t size)  
*create table*
- uint32\_t **getSize** () const  
*table buffers*
- Buffer **get** (uint32\_t index) const
- bool **isOwner** (uint32\_t index) const
- size\_t **getMemory** () const  
*memory usage*

#### 5.23.1 Detailed Description

[BufferTable](#) class

## 5.24 Tellusim::Tracing::BuildIndirect Struct Reference

build indirect parameters

```
#include <platform/TellusimTracing.h>
```

### Public Attributes

- uint32\_t **num\_primitives**
- uint32\_t **base\_primitive**
- uint32\_t **base\_vertex**
- uint32\_t **base\_transform**

### 5.24.1 Detailed Description

build indirect parameters

## 5.25 Tellusim::Canvas Class Reference

```
#include <interface/TellusimCanvas.h>
```

### Public Types

- using **CreateCallback** = Function< bool(const [Device](#) device, [Canvas](#) canvas, uint32\_t scale)>  
*create callback*
- using **PipelineCallback** = Function< bool([Pipeline](#) pipeline, [Canvas](#) canvas, [CanvasElement](#) element)>  
*pipeline callback*
- using **BeginCallback** = Function< bool([Command](#) command, [Canvas](#) canvas)>  
*begin callback*
- using **DrawCallback** = Function< bool([Command](#) command, [Canvas](#) canvas)>  
*draw callback*

### Public Member Functions

- **Canvas** ([Canvas](#) \*parent)
- void **clear** ()  
*clear canvas*
- bool **isCreated** () const  
*check canvas*
- uint32\_t **getScale** (const [Target](#) &target, uint32\_t scale=100) const  
*canvas scale*
- bool **create** (const [Device](#) &device, Format color, Format depth, uint32\_t multisample=1, uint32\_t scale=0)  
*create canvas*
- bool **create** (const [Device](#) &device, const [Target](#) &target, uint32\_t scale=0)
- void **setPipelineHash** (uint32\_t hash)  
*pipeline hash*
- uint32\_t **getPipelineHash** () const

- Format `getColorFormat ()` const  
*canvas parameters*
- Format `getDepthFormat ()` const
- `uint32_t getMultisample ()` const
- void `setOrder (int32_t order)`  
*canvas order*
- `int32_t getOrder ()` const
- void `setEnabled (bool enabled)`  
*canvas enabled flag*
- bool `isEnabled ()` const
- void `setViewport (const Viewport &viewport)`  
*canvas viewport*
- void `setViewport (uint32_t width, uint32_t height)`
- void `setViewport (float32_t width, float32_t height)`
- const `Viewport & getViewport ()` const
- `float32_t getWidth ()` const
- `float32_t getHeight ()` const
- void `clearColor ()`  
*canvas color*
- void `setColor (const Color &color)`
- void `setColor (float32_t r, float32_t g, float32_t b, float32_t a)`
- const `Color & getColor ()` const
- void `clearScissor ()`  
*canvas scissor*
- void `setScissor (const Rect &scissor)`
- const `Rect & getScissor ()` const
- void `clearTransform ()`  
*canvas transform*
- void `setTransform (const Matrix4x4f &transform)`
- const `Matrix4x4f & getTransform ()` const
- `uint32_t setParent (Canvas &parent)`  
*canvas parent*
- const `Canvas getParent ()` const
- `Canvas getParent ()`
- `uint32_t addChild (Canvas &child)`  
*canvas children*
- bool `removeChild (Canvas &child)`
- bool `raiseChild (Canvas &child)`
- bool `lowerChild (Canvas &child)`
- void `releaseChildren ()`
- `uint32_t findChild (const Canvas &child)` const
- bool `isChild (const Canvas &child)` const
- `uint32_t getNumChildren ()` const
- const `Array< Canvas > getChildren ()` const
- `Array< Canvas > getChildren ()`
- const `Canvas getChild (uint32_t index)` const
- `Canvas getChild (uint32_t index)`
- `uint32_t addElement (CanvasElement &element)`  
*canvas elements*
- bool `removeElement (CanvasElement &element)`
- bool `raiseElement (CanvasElement &element)`
- bool `lowerElement (CanvasElement &element)`
- `uint32_t findElement (const CanvasElement &element)` const

- bool **isElement** (const [CanvasElement](#) &element) const
- uint32\_t **getNumElements** () const
- const Array< [CanvasElement](#) > **getElements** () const
- Array< [CanvasElement](#) > **getElements** ()
- const [CanvasElement](#) **getElement** (uint32\_t index) const
- [CanvasElement](#) **getElement** (uint32\_t index)
- bool **isFont** (const char \*name) const
- canvas fonts*
- bool **addFont** (const char \*name, [Stream](#) &stream)
- bool **addFont** (const char \*name, const uint8\_t(\*blob)[256])
- void **removeFont** (const char \*name)
- [Font](#) **getFont** (const char \*name)
- bool **isTexture** (const char \*name) const
- canvas textures*
- bool **addTexture** (const char \*name, [Stream](#) &stream)
- bool **addTexture** (const char \*name, [Texture](#) &texture)
- bool **addTexture** (const char \*name, const uint8\_t(\*blob)[256])
- void **removeTexture** (const char \*name)
- [Texture](#) **getTexture** (const char \*name)
- void **setDepthMask** ([Pipeline::DepthMask](#) mask)
- depth parameters*
- [Pipeline::DepthMask](#) **getDepthMask** () const
- void **setDepthFunc** ([Pipeline::DepthFunc](#) func)
- [Pipeline::DepthFunc](#) **getDepthFunc** () const
- void **draw** ([Command](#) &command, const [Target](#) &target)
- draw canvas*
- void **draw** ([Command](#) &command)
- void **setCreateCallback** (const [CreateCallback](#) &func)
- [CreateCallback](#) **getCreateCallback** () const
- void **setPipelineCallback** (const [PipelineCallback](#) &func)
- [PipelineCallback](#) **getPipelineCallback** () const
- void **setBeginCallback** (const [BeginCallback](#) &func)
- [BeginCallback](#) **getBeginCallback** () const
- void **setDrawCallback** (const [DrawCallback](#) &func)
- [DrawCallback](#) **getDrawCallback** () const
- uint32\_t **getNumDrawPipelines** () const
- draw statistics*
- uint32\_t **getNumDrawElements** () const
- uint32\_t **getNumDrawCommands** () const
- [Rect](#) **getRect** () const

*canvas rectangle*

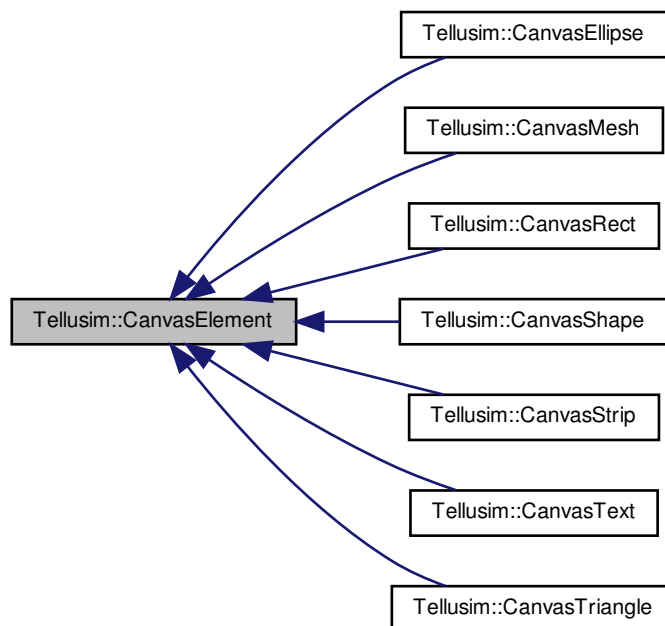
### 5.25.1 Detailed Description

[Canvas](#) class

## 5.26 Tellusim::CanvasElement Class Reference

```
#include <interface/TellusimCanvas.h>
```

Inheritance diagram for Tellusim::CanvasElement:



### Public Types

- enum [Mode](#) {  
**ModeSolid** = 0,  
**ModeTexture**,  
**ModeTextureFetch**,  
**ModeTextureClamp**,  
**ModeTextureCubic**,  
**ModeTextureCubic3x3**,  
**ModeTextureCubic5x5**,  
**ModeTextureRed**,  
**ModeTextureGreen**,  
**ModeTextureBlue**,  
**ModeTextureAlpha**,  
**ModeGradient**,  
**NumModes** }  
*Element modes.*
- enum [Align](#) {  
**AlignNone** = 0,  
**AlignLeft** = (1 << 0),  
**AlignRight** = (1 << 1),  
**AlignBottom** = (1 << 2),



```

AlignTop = (1 << 3),
AlignCenterX = (1 << 4),
AlignCenterY = (1 << 5),
AlignLeftBottom = (AlignLeft | AlignBottom),
AlignLeftTop = (AlignLeft | AlignTop),
AlignRightBottom = (AlignRight | AlignBottom),
AlignRightTop = (AlignRight | AlignTop),
AlignCenter = (AlignCenterX | AlignCenterY),
NumAligns = 6 }

```

*Element alignments.*

- enum **Stack** {  
**StackNone** = 0,  
**StackPush** = (1 << 0),  
**StackPop** = (1 << 1),  
**StackSet** = (1 << 2),  
**StackMul** = (1 << 3),  
**StackGet** = (1 << 4) }

*Element stack operations.*

- using **DrawCallback** = Function< bool(**Command** command, **CanvasElement** element)>  
*draw callback*

## Public Member Functions

- Type **getType** () const  
*element type*
- const char \* **getTypeName** () const
- bool **isText** () const
- bool **isMesh** () const
- bool **isRect** () const
- bool **isTriangle** () const
- bool **isEllipse** () const
- bool **isShape** () const
- bool **isStrip** () const
- void **setCanvas** (**Canvas** &canvas)  
*element canvas*
- const **Canvas** **getCanvas** () const
- **Canvas** **getCanvas** ()
- void **setMode** (**Mode** mode)  
*element mode*
- **Mode** **getMode** () const
- void **setAlign** (**Align** align)  
*element align*
- **Align** **getAlign** () const
- bool **hasAlign** (**Align** align) const
- bool **hasAligns** (**Align** aligns) const
- void **setOrder** (int32\_t order)  
*element order*
- int32\_t **getOrder** () const
- void **setEnabled** (bool enabled)  
*element enabled flag*
- bool **isEnabled** () const
- void **clearColor** ()  
*element color*

- void **setColor** (*Stack* op)
- void **setColor** (const *Color* &color, *Stack* op=StackNone)
- void **setColor** (float32\_t r, float32\_t g, float32\_t b, float32\_t a, *Stack* op=StackNone)
- const *Color* & **getColor** () const
- *Stack* **getColorOp** () const
- void **clearTransform** ()
  - element transform*
- void **setTransform** (*Stack* op)
- void **setTransform** (const *Matrix4x4f* &transform, *Stack* op=StackNone)
- const *Matrix4x4f* & **getTransform** () const
- *Stack* **getTransformOp** () const
- void **clearScissor** ()
  - element scissor*
- void **setScissor** (*Stack* op)
- void **setScissor** (const *Rect* &scissor, *Stack* op=StackNone)
- const *Rect* & **getScissor** () const
- *Stack* **getScissorOp** () const
- void **setMipmap** (float32\_t mipmap)
  - element mipmap number*
- float32\_t **getMipmap** () const
- void **setSampler** (*Sampler* &sampler)
  - sampler pointer*
- *Sampler* **getSampler** () const
- void **setFilter** (*Sampler::Filter* filter)
  - filter mode*
- *Sampler::Filter* **getFilter** () const
- void **setAnisotropy** (uint32\_t anisotropy)
- uint32\_t **getAnisotropy** () const
- void **setWrapMode** (*Sampler::WrapMode* mode)
  - wrapping mode*
- *Sampler::WrapMode* **getWrapMode** () const
- void **setTexture** (*Texture* &texture, bool linear=false)
  - texture pointer*
- *Texture* **getTexture** () const
- bool **getTextureLinear** () const
- void **setPipeline** (*Pipeline* pipeline)
  - pipeline pointer*
- *Pipeline* **getPipeline** () const
- void **setPrimitive** (*Pipeline::Primitive* primitive)
  - rasterization parameters*
- *Pipeline::Primitive* **getPrimitive** () const
- void **setCullMode** (*Pipeline::CullMode* mode)
- *Pipeline::CullMode* **getCullMode** () const
- void **setFrontMode** (*Pipeline::FrontMode* mode)
- *Pipeline::FrontMode* **getFrontMode** () const
- void **setBlend** (*Pipeline::BlendOp* op, *Pipeline::BlendFunc* src, *Pipeline::BlendFunc* dest)
  - blending parameters*
- *Pipeline::BlendOp* **getBlendOp** () const
- *Pipeline::BlendFunc* **getBlendSrcFunc** () const
- *Pipeline::BlendFunc* **getBlendDestFunc** () const
- void **setColorMask** (*Pipeline::ColorMask* mask)
  - color parameters*
- *Pipeline::ColorMask* **getColorMask** () const

- void **setDepthMask** ([Pipeline::DepthMask](#) mask)  
*depth parameters*
- [Pipeline::DepthMask](#) **getDepthMask** () const
- void **setDepthFunc** ([Pipeline::DepthFunc](#) func)
- [Pipeline::DepthFunc](#) **getDepthFunc** () const
- void **setStencilRef** (uint32\_t ref)  
*stencil parameters*
- void **setStencilFunc** ([Pipeline::StencilFunc](#) func, [Pipeline::StencilOp](#) fail\_op, [Pipeline::StencilOp](#) dfail\_op, [Pipeline::StencilOp](#) dpass\_op)
- uint32\_t **getStencilRef** () const
- [Pipeline::StencilFunc](#) **getStencilFunc** () const
- [Pipeline::StencilOp](#) **getStencilFailOp** () const
- [Pipeline::StencilOp](#) **getStencilDepthFailOp** () const
- [Pipeline::StencilOp](#) **getStencilDepthPassOp** () const
- void **setDrawCallback** (const [DrawCallback](#) &func)
- [DrawCallback](#) **getDrawCallback** () const
- const [Rect](#) & **getRect** ()  
*element rectangle*

#### Static Public Member Functions

- static const char \* **getTypeName** (Type type)

#### Friends

- class **Canvas**

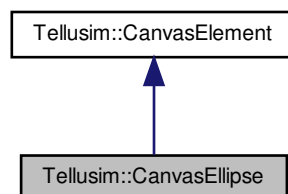
#### 5.26.1 Detailed Description

[CanvasElement](#) base class

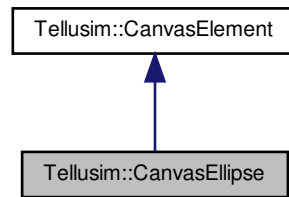
## 5.27 Tellusim::CanvasEllipse Class Reference

```
#include <interface/TellusimCanvas.h>
```

Inheritance diagram for Tellusim::CanvasEllipse:



Collaboration diagram for Tellusim::CanvasEllipse:



#### Public Member Functions

- **CanvasEllipse** ([Canvas](#) &canvas)
- **CanvasEllipse** ([Canvas](#) &canvas, float32\_t radius)
- void **setRadius** (float32\_t radius)  
*ellipse radius*
- float32\_t **getRadius** () const
- void **setTextureName** (const char \*name)  
*texture name*
- void **setTextureName** (const [String](#) &name)
- [String](#) **getTextureName** () const
- void **setStrokeColor** (const [Color](#) &color)  
*stroke color*
- const [Color](#) & **getStrokeColor** () const
- void **setStrokeStyle** (const [StrokeStyle](#) &style)  
*stroke style*
- const [StrokeStyle](#) & **getStrokeStyleConst** () const
- const [StrokeStyle](#) & **getStrokeStyle** () const
- [StrokeStyle](#) & **getStrokeStyle** ()
- void **setGradientStyle** (const [GradientStyle](#) &style)  
*gradient style*
- const [GradientStyle](#) & **getGradientStyleConst** () const
- const [GradientStyle](#) & **getGradientStyle** () const
- [GradientStyle](#) & **getGradientStyle** ()
- void **setPosition** (const [Vector3f](#) &position)  
*ellipse positions*
- void **setPosition0** (const [Vector3f](#) &position)
- void **setPosition1** (const [Vector3f](#) &position)
- void **setPosition** (float32\_t x, float32\_t y, float32\_t z=0.0f)
- void **setPosition0** (float32\_t x, float32\_t y, float32\_t z=0.0f)
- void **setPosition1** (float32\_t x, float32\_t y, float32\_t z=0.0f)
- void **setPosition** (const [Vector3f](#) &position\_0, const [Vector3f](#) &position\_1)
- const [Vector3f](#) & **getPosition0** () const
- const [Vector3f](#) & **getPosition1** () const
- void **setTexCoord** (const [Rect](#) &texcoord)  
*texture coordinates*
- void **setTexCoord** (float32\_t left, float32\_t right, float32\_t bottom, float32\_t top)
- const [Rect](#) & **getTexCoord** () const

## Additional Inherited Members

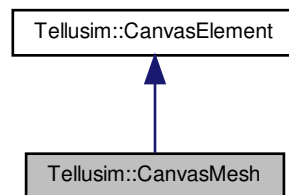
## 5.27.1 Detailed Description

[CanvasEllipse](#) class

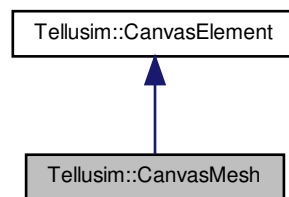
## 5.28 Tellusim::CanvasMesh Class Reference

```
#include <interface/TellusimCanvas.h>
```

Inheritance diagram for Tellusim::CanvasMesh:



Collaboration diagram for Tellusim::CanvasMesh:



## Public Member Functions

- **CanvasMesh** ([Canvas](#) &canvas)
- **CanvasMesh** ([Canvas](#) &canvas, [Mode](#) mode)
- void **setTextureName** (const char \*name)  
*texture name*
- void **setTextureName** (const [String](#) &name)
- [String](#) **getTextureName** () const
- void **setGradientStyle** (const [GradientStyle](#) &style)

*gradient style*

- const [GradientStyle](#) & **getGradientStyleConst** () const
- const [GradientStyle](#) & **getGradientStyle** () const
- [GradientStyle](#) & **getGradientStyle** ()
- void **clearVertices** ()

*mesh vertices*

- void **setNumVertices** (uint32\_t num\_vertices)
- void **reserveVertices** (uint32\_t num\_vertices)
- uint32\_t **getNumVertices** () const
- void **setVertices** (const [CanvasVertex](#) \*vertices, uint32\_t num\_vertices)
- void **addVertices** (const [CanvasVertex](#) \*vertices, uint32\_t num\_vertices)
- const [CanvasVertex](#) \* **getVertices** () const
- [CanvasVertex](#) \* **getVertices** ()
- void **setVertex** (uint32\_t index, const [CanvasVertex](#) &vertex)
- const [CanvasVertex](#) & **getVertex** (uint32\_t index) const
- [CanvasVertex](#) & **getVertex** (uint32\_t index)
- void **setVertexPosition** (uint32\_t index, const [Vector3f](#) &position)

*vertex positions*

- void **setVertexPosition** (uint32\_t index, float32\_t x, float32\_t y, float32\_t z=0.0f)
- [Vector3f](#) **getVertexPosition** (uint32\_t index) const
- void **setVertexTexCoord** (uint32\_t index, const [Vector2f](#) &texcoord)

*vertex texture coordinates*

- void **setVertexTexCoord** (uint32\_t index, float32\_t s, float32\_t t)
- [Vector2f](#) **getVertexTexCoord** (uint32\_t index) const
- void **setVertexColor** (uint32\_t index, const [Color](#) &color)

*vertex colors*

- void **setVertexColor** (uint32\_t index, uint32\_t color)
- uint32\_t **getVertexColor** (uint32\_t index) const
- uint32\_t **addVertex** (const [Vector3f](#) &position)

*add mesh vertex*

- uint32\_t **addVertex** (const [Vector3f](#) &position, uint32\_t color)
- uint32\_t **addVertex** (const [Vector3f](#) &position, const [Vector2f](#) &texcoord)
- uint32\_t **addVertex** (const [Vector3f](#) &position, const [Vector2f](#) &texcoord, uint32\_t color)
- uint32\_t **addVertex** (float32\_t x, float32\_t y, float32\_t z, float32\_t s, float32\_t t, uint32\_t color=0xffffffffu)
- uint32\_t **addVertex** (float32\_t x, float32\_t y, float32\_t z, uint32\_t color=0xffffffffu)
- uint32\_t **addVertex** (float32\_t x, float32\_t y, uint32\_t color=0xffffffffu)
- void **clearIndices** ()

*mesh indices*

- void **setNumIndices** (uint32\_t num\_indices)
- void **reserveIndices** (uint32\_t num\_indices)
- uint32\_t **getNumIndices** () const
- void **setIndices** (const uint32\_t \*indices, uint32\_t num\_indices)
- void **addIndices** (const uint32\_t \*indices, uint32\_t num\_indices)
- const uint32\_t \* **getIndices** () const
- uint32\_t \* **getIndices** ()
- void **setIndex** (uint32\_t index, uint32\_t value)
- uint32\_t **getIndex** (uint32\_t index) const
- void **addIndex** (uint32\_t i0)

*add mesh indices*

- void **addIndices** (uint32\_t i0, uint32\_t i1)
- void **addIndices** (uint32\_t i0, uint32\_t i1, uint32\_t i2)
- void **addIndices** (uint32\_t i0, uint32\_t i1, uint32\_t i2, uint32\_t i3)
- void **setRect** (const [Rect](#) &rect)

*mesh rectangle*

## Additional Inherited Members

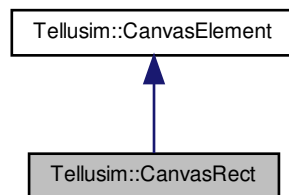
## 5.28.1 Detailed Description

[CanvasMesh](#) class

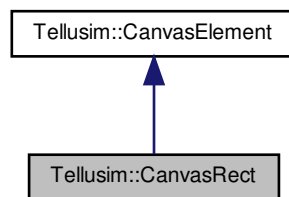
## 5.29 Tellusim::CanvasRect Class Reference

```
#include <interface/TellusimCanvas.h>
```

Inheritance diagram for Tellusim::CanvasRect:



Collaboration diagram for Tellusim::CanvasRect:



## Public Member Functions

- **CanvasRect** ([Canvas](#) &canvas)
- **CanvasRect** ([Canvas](#) &canvas, float32\_t radius)
- **CanvasRect** ([Canvas](#) &canvas, float32\_t radius, const [Vector2f](#) &size)
- void [setRadius](#) (float32\_t radius)  
    *rect radius*
- float32\_t [getRadius](#) () const
- void [setTextureName](#) (const char \*name)

- texture name*
  - void **setTextureName** (const [String](#) &name)
  - [String](#) **getTextureName** () const
  - void **setStrokeColor** (const [Color](#) &color)
- stroke color*
  - const [Color](#) & **getStrokeColor** ()
  - void **setStrokeStyle** (const [StrokeStyle](#) &style)
- stroke style*
  - const [StrokeStyle](#) & **getStrokeStyleConst** () const
  - const [StrokeStyle](#) & **getStrokeStyle** () const
  - [StrokeStyle](#) & **getStrokeStyle** ()
  - void **setGradientStyle** (const [GradientStyle](#) &style)
- gradient style*
  - const [GradientStyle](#) & **getGradientStyleConst** () const
  - const [GradientStyle](#) & **getGradientStyle** () const
  - [GradientStyle](#) & **getGradientStyle** ()
  - void **setSize** (const [Vector2f](#) &size)
- rect size*
  - void **setSize** (float32\_t width, float32\_t height)
  - const [Vector2f](#) & **getSize** () const
  - float32\_t **getWidth** () const
  - float32\_t **getHeight** () const
  - void **setPosition** (const [Vector3f](#) &position)
- rect position*
  - void **setPosition** (float32\_t x, float32\_t y, float32\_t z=0.0f)
  - const [Vector3f](#) & **getPosition** () const
  - void **setTexCoord** (const [Rect](#) &texcoord)
- texture coordinates*
  - void **setTexCoord** (float32\_t left, float32\_t right, float32\_t bottom, float32\_t top)
  - const [Rect](#) & **getTexCoord** () const

## Additional Inherited Members

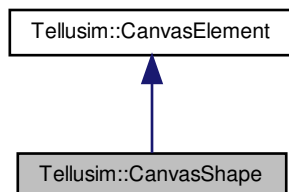
### 5.29.1 Detailed Description

[CanvasRect](#) class

## 5.30 Tellusim::CanvasShape Class Reference

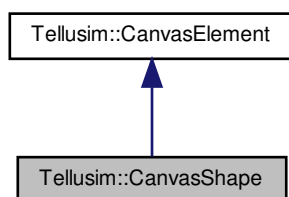
```
#include <interface/TellusimCanvas.h>
```

Inheritance diagram for Tellusim::CanvasShape:





Collaboration diagram for Tellusim::CanvasShape:



#### Public Member Functions

- **CanvasShape** ([Canvas](#) &canvas)
- **CanvasShape** ([Canvas](#) &canvas, bool cubic)
- void **setCubic** (bool cubic)  
*cubic flag*
- bool **isCubic** () const
- void **setThreshold** (float32\_t threshold)  
*cubic to quadratic threshold*
- float32\_t **getThreshold** () const
- void **setStrokeColor** (const [Color](#) &color)  
*stroke color*
- const [Color](#) & **getStrokeColor** () const
- void **setStrokeStyle** (const [StrokeStyle](#) &style)  
*stroke style*
- const [StrokeStyle](#) & **getStrokeStyleConst** () const
- const [StrokeStyle](#) & **getStrokeStyle** () const
- [StrokeStyle](#) & **getStrokeStyle** ()
- void **setGradientStyle** (const [GradientStyle](#) &style)  
*gradient style*
- const [GradientStyle](#) & **getGradientStyleConst** () const
- const [GradientStyle](#) & **getGradientStyle** () const
- [GradientStyle](#) & **getGradientStyle** ()
- bool **createSVG** (const char \*src, float32\_t scale=1.0f)  
*create shape from SVG path*
- void **clearPositions** ()  
*shape positions*
- void **setNumPositions** (uint32\_t num\_positions)
- void **reservePositions** (uint32\_t num\_positions)
- uint32\_t **getNumPositions** () const
- void **setPositions** (const [Vector3f](#) \*positions, uint32\_t num\_positions)
- void **addPositions** (const [Vector3f](#) \*positions, uint32\_t num\_positions)
- const [Vector3f](#) \* **getPositions** () const
- [Vector3f](#) \* **getPositions** ()
- void **setPosition** (uint32\_t index, const [Vector3f](#) &position)
- void **setPosition** (uint32\_t index, float32\_t x, float32\_t y, float32\_t z=0.0f)
- const [Vector3f](#) & **getPosition** (uint32\_t index) const

- [Vector3f](#) & **getPosition** (uint32\_t index)
- uint32\_t **addPosition** (const [Vector2f](#) &position)  
*add shape position*
- uint32\_t **addPosition** (const [Vector3f](#) &position)
- uint32\_t **addPosition** (float32\_t x, float32\_t y, float32\_t z=0.0f)
- void **setTexCoord** (const [Rect](#) &texcoord)  
*texture coordinates*
- void **setTexCoord** (float32\_t left, float32\_t right, float32\_t bottom, float32\_t top)
- const [Rect](#) & **getTexCoord** () const

## Additional Inherited Members

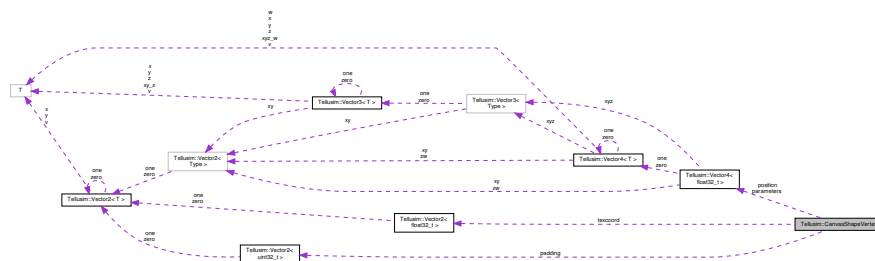
### 5.30.1 Detailed Description

[CanvasShape](#) class

## 5.31 Tellusim::CanvasShapeVertex Struct Reference

```
#include <interface/TellusimCanvas.h>
```

Collaboration diagram for Tellusim::CanvasShapeVertex:



## Public Attributes

- [Vector4f](#) **position**
- [Vector4f](#) **parameters**
- [Vector2f](#) **texcoord**
- [Vector2u](#) **padding**

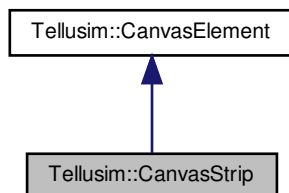
### 5.31.1 Detailed Description

[Canvas](#) shape vertex

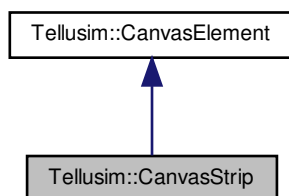
## 5.32 Tellusim::CanvasStrip Class Reference

```
#include <interface/TellusimCanvas.h>
```

Inheritance diagram for Tellusim::CanvasStrip:



Collaboration diagram for Tellusim::CanvasStrip:



## Public Member Functions

- **CanvasStrip** ([Canvas](#) &canvas)
- **CanvasStrip** ([Canvas](#) &canvas, float32\_t width)
- void [setWidth](#) (float32\_t width)  
*strip width*
- float32\_t **getWidth** () const
- void [setOffset](#) (float32\_t offset)  
*strip offset*
- float32\_t **getOffset** () const
- void [setStrokeColor](#) (const [Color](#) &color)  
*stroke color*
- const [Color](#) & **getStrokeColor** () const
- void [setStrokeStyle](#) (const [StrokeStyle](#) &style)  
*stroke style*
- const [StrokeStyle](#) & **getStrokeStyleConst** () const
- const [StrokeStyle](#) & **getStrokeStyle** () const

- [StrokeStyle](#) & **getStrokeStyle** ()
- void **createQuadratic** (const [Vector2f](#) &p0, const [Vector2f](#) &p1, const [Vector2f](#) &p2, float32\_t threshold=1.0f)  
*create quadratic spline*
- void **createQuadratic** (const [Vector3f](#) &p0, const [Vector3f](#) &p1, const [Vector3f](#) &p2, float32\_t threshold=1.0f)
- void **createCubic** (const [Vector2f](#) &p0, const [Vector2f](#) &p1, const [Vector2f](#) &p2, const [Vector2f](#) &p3, float32\_t threshold=1.0f)  
*create cubic spline*
- void **createCubic** (const [Vector3f](#) &p0, const [Vector3f](#) &p1, const [Vector3f](#) &p2, const [Vector3f](#) &p3, float32\_t threshold=1.0f)
- void **clearPositions** ()  
*strip positions*
- void **setNumPositions** (uint32\_t num\_positions)
- void **reservePositions** (uint32\_t num\_positions)
- uint32\_t **getNumPositions** () const
- void **setPositions** (const [Vector3f](#) \*positions, uint32\_t num\_positions)
- void **addPositions** (const [Vector3f](#) \*positions, uint32\_t num\_positions)
- const [Vector3f](#) \* **getPositions** () const
- [Vector3f](#) \* **getPositions** ()
- void **setPosition** (uint32\_t index, const [Vector3f](#) &position)
- void **setPosition** (uint32\_t index, float32\_t x, float32\_t y, float32\_t z=0.0f)
- const [Vector3f](#) & **getPosition** (uint32\_t index) const
- [Vector3f](#) & **getPosition** (uint32\_t index)
- uint32\_t **addPosition** (const [Vector2f](#) &position)  
*add strip position*
- uint32\_t **addPosition** (const [Vector3f](#) &position)
- uint32\_t **addPosition** (float32\_t x, float32\_t y, float32\_t z=0.0f)

## Additional Inherited Members

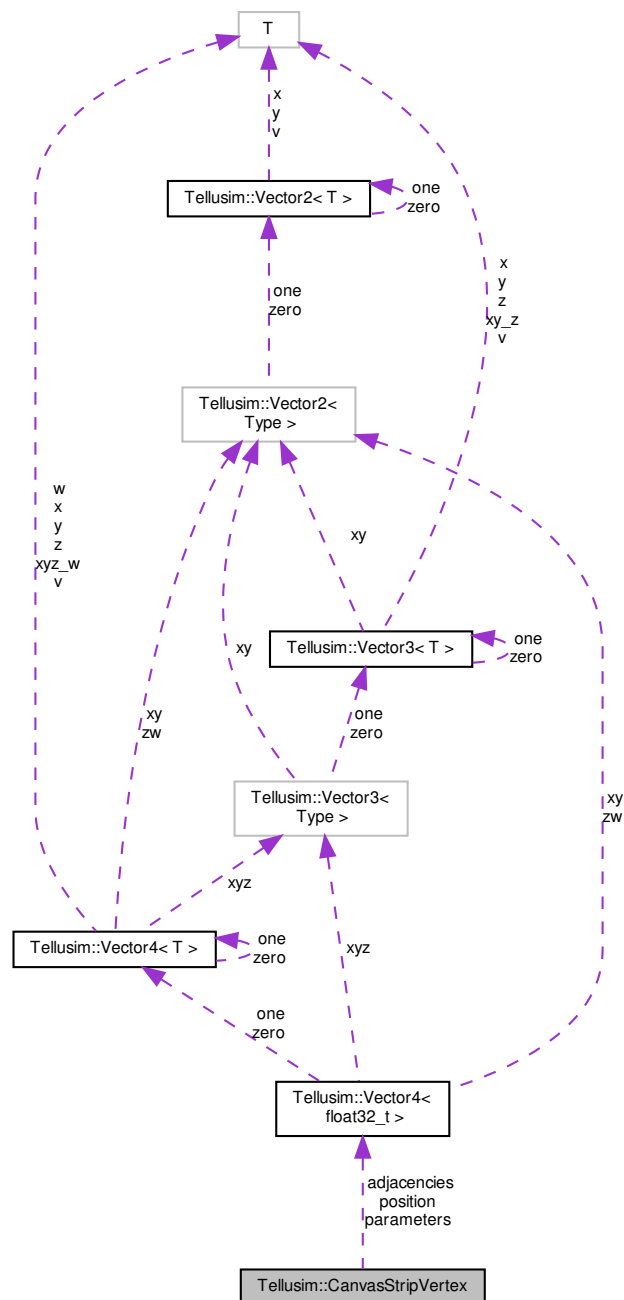
### 5.32.1 Detailed Description

[CanvasStrip](#) class

## 5.33 Tellusim::CanvasStripVertex Struct Reference

```
#include <interface/TellusimCanvas.h>
```

Collaboration diagram for Tellusim::CanvasStripVertex:



## Public Attributes

- **Vector4f position**
- **Vector4f parameters**
- **Vector4f adjacencies**

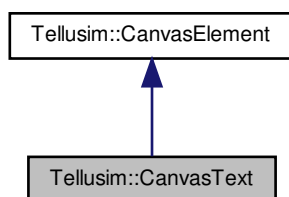
### 5.33.1 Detailed Description

[Canvas](#) strip vertex

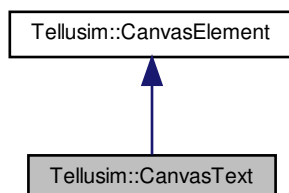
## 5.34 Tellusim::CanvasText Class Reference

```
#include <interface/TellusimCanvas.h>
```

Inheritance diagram for Tellusim::CanvasText:



Collaboration diagram for Tellusim::CanvasText:



### Public Member Functions

- **CanvasText** ([Canvas](#) &canvas)
- **CanvasText** ([Canvas](#) &canvas, const char \*text)
- **CanvasText** ([Canvas](#) &canvas, const [String](#) &text)
- void [setFontName](#) (const char \*name)  
*font name*
- void **setFontName** (const [String](#) &name)
- [String](#) **getFontName** () const
- void [setFontColor](#) (const [Color](#) &color)  
*font color*

- const [Color](#) & **getFontColor** () const
- bool **setFontSize** (uint32\_t scale)
- font style*
- uint32\_t **getFontSize** () const
- bool **setFontScale** (uint32\_t scale)
- uint32\_t **getFontScale** () const
- bool **setFontStyle** (const [FontStyle](#) &style)
- const [FontStyle](#) & **getFontStyleConst** () const
- const [FontStyle](#) & **getFontStyle** () const
- [FontStyle](#) & **getFontStyle** ()
- void **setPosition** (const [Vector3f](#) &position)
- font position*
- void **setPosition** (float32\_t x, float32\_t y, float32\_t z=0.0f)
- const [Vector3f](#) & **getPosition** () const
- void **setText** (const char \*text)
- font text*
- void **setText** (const [String](#) &text)
- [String](#) **getText** () const
- void **clearBatches** ()
- font batches*
- void **setBatches** (const Array< [FontBatch](#) > &batches)
- void **setBatches** (const [FontBatch](#) \*batches, uint32\_t num\_batches)

#### Additional Inherited Members

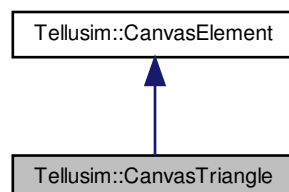
##### 5.34.1 Detailed Description

[CanvasText](#) class

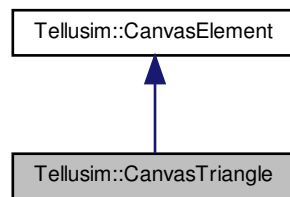
## 5.35 Tellusim::CanvasTriangle Class Reference

```
#include <interface/TellusimCanvas.h>
```

Inheritance diagram for Tellusim::CanvasTriangle:



Collaboration diagram for Tellusim::CanvasTriangle:



#### Public Member Functions

- **CanvasTriangle** ([Canvas](#) &canvas)
- **CanvasTriangle** ([Canvas](#) &canvas, float32\_t radius)
- void [setRadius](#) (float32\_t radius)  
*triangle radius*
- float32\_t **getRadius** () const
- void [setStrokeColor](#) (const [Color](#) &color)  
*stroke color*
- const [Color](#) & **getStrokeColor** () const
- void [setStrokeStyle](#) (const [StrokeStyle](#) &style)  
*stroke style*
- const [StrokeStyle](#) & **getStrokeStyleConst** () const
- const [StrokeStyle](#) & **getStrokeStyle** () const
- [StrokeStyle](#) & **getStrokeStyle** ()
- void [setGradientStyle](#) (const [GradientStyle](#) &style)  
*gradient style*
- const [GradientStyle](#) & **getGradientStyleConst** () const
- const [GradientStyle](#) & **getGradientStyle** () const
- [GradientStyle](#) & **getGradientStyle** ()
- void [setPosition0](#) (const [Vector3f](#) &position)  
*triangle positions*
- void [setPosition1](#) (const [Vector3f](#) &position)
- void [setPosition2](#) (const [Vector3f](#) &position)
- void [setPosition0](#) (float32\_t x, float32\_t y, float32\_t z=0.0f)
- void [setPosition1](#) (float32\_t x, float32\_t y, float32\_t z=0.0f)
- void [setPosition2](#) (float32\_t x, float32\_t y, float32\_t z=0.0f)
- void [setPosition](#) (const [Vector3f](#) &position\_0, const [Vector3f](#) &position\_1, const [Vector3f](#) &position\_2)
- const [Vector3f](#) & **getPosition0** () const
- const [Vector3f](#) & **getPosition1** () const
- const [Vector3f](#) & **getPosition2** () const

#### Additional Inherited Members

##### 5.35.1 Detailed Description

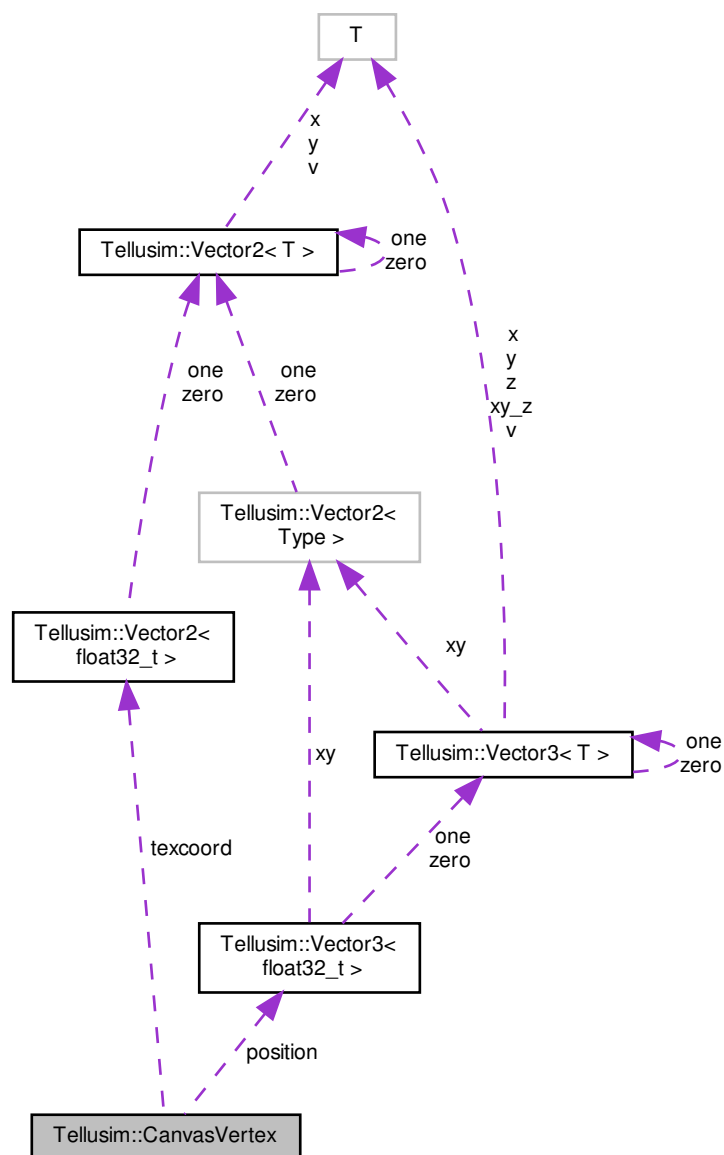
[CanvasTriangle](#) class



### 5.36 Tellusim::CanvasVertex Struct Reference

```
#include <interface/TellusimCanvas.h>
```

Collaboration diagram for Tellusim::CanvasVertex:



## Public Attributes

- **Vector3f** position
- **Vector2f** texcoord
- **uint32\_t** color

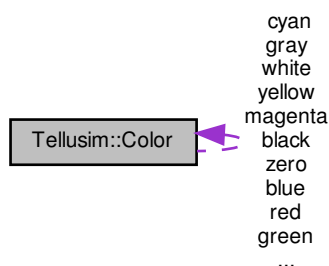
### 5.36.1 Detailed Description

[Canvas](#) vertex

## 5.37 Tellusim::Color Struct Reference

```
#include <math/TellusimColor.h>
```

Collaboration diagram for Tellusim::Color:



### Public Types

- enum { **Size** = 4 }

### Public Member Functions

- **Color** (const char \*src)
- **Color** (uint32\_t color)
- **Color** (float32\_t c)
- **Color** (float32\_t l, float32\_t a)
- **Color** (const [Color](#) &c, float32\_t a)
- **Color** (const float32\_t \*c, uint32\_t size=[Size](#))
- **Color** (uint32\_t r, uint32\_t g, uint32\_t b, uint32\_t a=255)
- **Color** (float32\_t r, float32\_t g, float32\_t b, float32\_t a=1.0f)
- void [set](#) (const [Color](#) &c, float32\_t A)
  - update color data*
- void **set** (float32\_t R, float32\_t G, float32\_t B, float32\_t A)
- void **set** (const float32\_t \*1 c, uint32\_t size=[Size](#))
- void **get** (float32\_t \*1 c, uint32\_t size=[Size](#)) const
- bool [isValid](#) () const
  - check color*
- **operator bool** () const
- bool [isBlack](#) () const
  - color parameters*
- bool **isWhite** () const

- bool **isTransparent** () const
- bool **isOpaque** () const
- **Color** & **operator\*=** (float32\_t l)
- color operators*
- **Color** & **operator/=** (float32\_t l)
- **Color** & **operator+=** (float32\_t l)
- **Color** & **operator-=** (float32\_t l)
- **Color** & **operator\*=** (const **Color** &c)
- **Color** & **operator/=** (const **Color** &c)
- **Color** & **operator+=** (const **Color** &c)
- **Color** & **operator-=** (const **Color** &c)
- void **gammaToLinear** (float32\_t \*1 v) const
- gamma format*
- void **linearToGamma** (float32\_t \*1 v) const
- **Color** **gammaToLinear** () const
- **Color** **linearToGamma** () const
- void **sRGBtoLinear** (float32\_t \*1 v) const
- sRGB format*
- void **linearToSRGB** (float32\_t \*1 v) const
- **Color** **sRGBtoLinear** () const
- **Color** **linearToSRGB** () const
- void **setRGBAu8** (uint32\_t R, uint32\_t G, uint32\_t B, uint32\_t A)
- RGBAu8 color components.*
- uint8\_t **getRu8** () const
- uint8\_t **getGu8** () const
- uint8\_t **getBu8** () const
- uint8\_t **getAu8** () const
- void **setRGBAu8** (uint32\_t color)
- RGBAu8 color format.*
- void **setBGR Au8** (uint32\_t color)
- void **setABGR Au8** (uint32\_t color)
- uint32\_t **getRGB Au8** () const
- uint32\_t **getBGR Au8** () const
- uint32\_t **getABGR Au8** () const
- bool **setHTML** (const char \*src)
- HTML color format.*
- void **setTemperature** (float32\_t t)
- temperature in K*
- const float32\_t & **operator[]** (uint32\_t index) const
- Color data.*
- float32\_t & **operator[]** (uint32\_t index)

#### Static Public Member Functions

- static **Color** **temperature** (float32\_t t)

## Public Attributes

```

•
union {
    struct {
        float32_t r
        float32_t g
        float32_t b
        float32_t a
    }
    float32_t c [Size]
};

```

## Static Public Attributes

- static const [Color](#) **zero**  
*default colors*
- static const [Color](#) **black**
- static const [Color](#) **white**
- static const [Color](#) **gray**
- static const [Color](#) **red**  
*rgb colors*
- static const [Color](#) **yellow**
- static const [Color](#) **green**
- static const [Color](#) **cyan**
- static const [Color](#) **blue**
- static const [Color](#) **magenta**

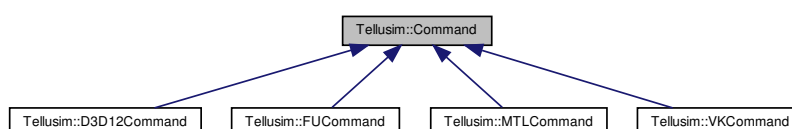
## 5.37.1 Detailed Description

[Color](#) class

## 5.38 Tellusim::Command Class Reference

```
#include <platform/TellusimCommand.h>
```

Inheritance diagram for Tellusim::Command:



## Classes

- struct [DrawArraysIndirect](#)  
*draw arrays indirect parameters*
- struct [DrawElementsIndirect](#)  
*draw elements indirect parameters*
- struct [DrawMeshIndirect](#)  
*draw mesh indirect parameters*

## Public Member Functions

- Platform [getPlatform](#) () const  
*command platform*
- const char \* [getPlatformName](#) () const
- uint32\_t [getIndex](#) () const  
*command device index*
- void [setPipeline](#) ([Pipeline](#) &pipeline)  
*set pipeline*
- [Pipeline](#) [getPipeline](#) () const
- void [setViewport](#) (uint32\_t index, const [Viewport](#) &viewport)  
*set viewport parameters*
- void [setViewports](#) (const [Viewport](#) \*viewports, uint32\_t num\_viewports)
- void [setScissor](#) (uint32\_t index, const [Scissor](#) &scissor)  
*set scissor parameters*
- void [setScissors](#) (const [Scissor](#) \*scissors, uint32\_t num\_scissors)
- void [setSampler](#) (uint32\_t index, [Sampler](#) &sampler)  
*set samplers*
- void [setSamplers](#) (uint32\_t index, const Array< [Sampler](#) > &samplers)
- void [setSamplers](#) (uint32\_t index, const InitializerList< [Sampler](#) > &samplers)
- void [setTexture](#) (uint32\_t index, [Texture](#) &texture)  
*set textures*
- void [setTexture](#) (uint32\_t index, [Texture](#) &texture, const [Slice](#) &slice)
- void [setTextures](#) (uint32\_t index, const Array< [Texture](#) > &textures)
- void [setTextures](#) (uint32\_t index, const InitializerList< [Texture](#) > &textures)
- void [setSurfaceTexture](#) (uint32\_t index, [Texture](#) &texture)  
*set surfaces*
- void [setSurfaceTexture](#) (uint32\_t index, [Texture](#) &texture, const [Slice](#) &slice, Format format=FormatUnknown)
- void [setSurfaceTextures](#) (uint32\_t index, const Array< [Texture](#) > &textures)
- void [setSurfaceTextures](#) (uint32\_t index, const InitializerList< [Texture](#) > &textures)
- void \* [getUniformData](#) (uint32\_t index, size\_t size)  
*set uniforms*
- void [setUniformData](#) (uint32\_t index, const void \*src, size\_t size)
- void [setUniformBuffer](#) (uint32\_t index, [Buffer](#) &buffer, size\_t offset=0, size\_t size=0)
- void [setUniformOffset](#) (uint32\_t index, size\_t offset, bool relative=false)
- void [setUniformBuffers](#) (uint32\_t index, const Array< [Buffer](#) > &buffers)
- void [setUniformBuffers](#) (uint32\_t index, const Array< [Buffer](#) > &buffers, const Array< size\_t > &offsets)
- void [setUniformBuffers](#) (uint32\_t index, const InitializerList< [Buffer](#) > &buffers)
- void [setUniformBuffers](#) (uint32\_t index, const InitializerList< [Buffer](#) > &buffers, const InitializerList< size\_t > &offsets)
- template<class Type >  
void [setUniform](#) (uint32\_t index, const Array< Type > &data)

- `template<class Type >`  
`void setUniform (uint32_t index, const Type &data)`
- `void * getStorageData (uint32_t index, size_t size)`  
*set storages*
- `void setStorageData (uint32_t index, const void *src, size_t size)`
- `void setStorageBuffer (uint32_t index, Buffer &buffer, size_t offset=0, size_t size=0)`
- `void setStorageOffset (uint32_t index, size_t offset, bool relative=false)`
- `void setStorageBuffers (uint32_t index, const Array< Buffer > &buffers)`
- `void setStorageBuffers (uint32_t index, const Array< Buffer > &buffers, const Array< size_t > &offsets)`
- `void setStorageBuffers (uint32_t index, const InitializerList< Buffer > &buffers)`
- `void setStorageBuffers (uint32_t index, const InitializerList< Buffer > &buffers, const InitializerList< size_t > &offsets)`
- `template<class Type >`  
`void setStorage (uint32_t index, const Array< Type > &data)`
- `template<class Type >`  
`void setStorage (uint32_t index, const Type &data)`
- `void setTracing (uint32_t index, Tracing &tracing)`  
*set tracings*
- `void setTracings (uint32_t index, const Array< Tracing > &tracings)`
- `void setTracings (uint32_t index, const InitializerList< Tracing > &tracings)`
- `void setTexelBuffer (uint32_t index, Buffer &buffer)`  
*set texel buffers*
- `void setTexelBuffers (uint32_t index, const Array< Buffer > &buffers)`
- `void setTexelBuffers (uint32_t index, const InitializerList< Buffer > &buffers)`
- `void setTextureTable (uint32_t index, TextureTable &table)`  
*set texture tables*
- `void setTextureTables (uint32_t index, const Array< TextureTable > &tables)`
- `void setTextureTables (uint32_t index, const InitializerList< TextureTable > &tables)`
- `void setStorageTable (uint32_t index, BufferTable &table)`  
*set storage tables*
- `void setStorageTables (uint32_t index, const Array< BufferTable > &tables)`
- `void setStorageTables (uint32_t index, const InitializerList< BufferTable > &tables)`
- `void * getVertexData (uint32_t index, size_t size)`  
*set vertices*
- `void setVertexData (uint32_t index, const void *src, size_t size)`
- `void setVertexBuffer (uint32_t index, Buffer &buffer, size_t offset=0)`
- `void setVertexOffset (uint32_t index, size_t offset, bool relative=false)`
- `void setVertexBuffers (uint32_t index, const Array< Buffer > &buffers)`
- `void setVertexBuffers (uint32_t index, const Array< Buffer > &buffers, const Array< size_t > &offsets)`
- `void setVertexBuffers (uint32_t index, const InitializerList< Buffer > &buffers)`
- `void setVertexBuffers (uint32_t index, const InitializerList< Buffer > &buffers, const InitializerList< size_t > &offsets)`
- `template<class Type >`  
`void setVertices (uint32_t index, const Array< Type > &vertices)`
- `template<class Type, size_t Size>`  
`void setVertices (uint32_t index, const Type(&vertices)[Size])`
- `template<class Type >`  
`void setVertices (uint32_t index, const InitializerList< Type > &vertices)`
- `void * getIndexData (Format format, size_t size)`  
*set indices*
- `void setIndexData (Format format, const void *src, size_t size)`
- `void setIndexBuffer (Format format, Buffer &buffer, size_t offset=0)`
- `void setIndexOffset (size_t offset, bool relative=false)`

- template<class Type >  
void **setIndices** (Format format, const Array< Type > &indices)
- template<class Type , size\_t Size>  
void **setIndices** (Format format, const Type(&indices)[Size])
- void **setIndices** (const InitializerList< uint16\_t > &indices)
- void \* **getIndirectData** (size\_t size)  
*set indirect*
- void **setIndirectData** (const void \*src, size\_t size)
- void **setIndirectBuffer** (Buffer &buffer, size\_t offset=0)
- void **setIndirectOffset** (size\_t offset, bool relative=false)
- template<class Type >  
void **setIndirect** (const Type &data)
- void **setBlendColor** (const Color &color)  
*blending parameters*
- void **setBlendColor** (float32\_t r, float32\_t g, float32\_t b, float32\_t a)
- void **setStencilRef** (uint32\_t ref)  
*stencil parameters*
- void **drawArrays** (uint32\_t num\_vertices, uint32\_t base\_vertex=0)  
*draw arrays*
- void **drawArraysInstanced** (uint32\_t num\_vertices, uint32\_t base\_vertex, uint32\_t num\_instances, uint32\_t base\_instance=0)
- void **drawArraysIndirect** (uint32\_t num\_draws, size\_t stride=sizeof(DrawArraysIndirect))
- void **drawArraysIndirect** (Buffer &buffer, size\_t offset, uint32\_t num\_draws, size\_t stride=sizeof(DrawArraysIndirect))
- void **drawElements** (uint32\_t num\_indices, uint32\_t base\_index=0, int32\_t base\_vertex=0)  
*draw elements*
- void **drawElementsInstanced** (uint32\_t num\_indices, uint32\_t base\_index, uint32\_t num\_instances)
- void **drawElementsInstanced** (uint32\_t num\_indices, uint32\_t base\_index, int32\_t base\_vertex, uint32\_t num\_instances, uint32\_t base\_instance=0)
- void **drawElementsIndirect** (uint32\_t num\_draws, size\_t stride=sizeof(DrawElementsIndirect))
- void **drawElementsIndirect** (Buffer &buffer, size\_t offset, uint32\_t num\_draws, size\_t stride=sizeof(DrawElementsIndirect))
- void **drawMesh** (uint32\_t width, uint32\_t height=1, uint32\_t depth=1)  
*draw mesh*
- void **drawMeshIndirect** (uint32\_t num\_draws, size\_t stride=sizeof(DrawMeshIndirect))
- void **drawMeshIndirect** (Buffer &buffer, size\_t offset, uint32\_t num\_draws, size\_t stride=sizeof(DrawMeshIndirect))
- void **beginConditional** (Buffer &buffer, size\_t offset)  
*begin/end conditional*
- void **endConditional** ()
- bool **beginQuery** (Query &query)  
*begin/end query*
- void **endQuery** (Query &query)

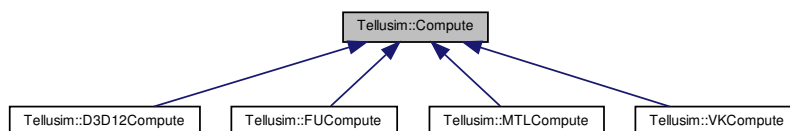
#### 5.38.1 Detailed Description

Command class

### 5.39 Tellusim::Compute Class Reference

```
#include <platform/TellusimCompute.h>
```

Inheritance diagram for Tellusim::Compute:



#### Classes

- struct [DispatchIndirect](#)  
*dispatch indirect parameters*

#### Public Member Functions

- Platform [getPlatform](#) () const  
*compute platform*
- const char \* [getPlatformName](#) () const
- uint32\_t [getIndex](#) () const  
*compute device index*
- void [setKernel](#) (Kernel &kernel)  
*set kernel*
- Kernel [getKernel](#) () const
- void [setTraversal](#) (Traversal &traversal)  
*set traversal*
- Traversal [getTraversal](#) () const
- void [setSampler](#) (uint32\_t index, Sampler &sampler)  
*set samplers*
- void [setSamplers](#) (uint32\_t index, const Array< Sampler > &samplers)
- void [setSamplers](#) (uint32\_t index, const InitializerList< Sampler > &samplers)
- void [setTexture](#) (uint32\_t index, Texture &texture)  
*set textures*
- void [setTexture](#) (uint32\_t index, Texture &texture, const Slice &slice)
- void [setTextures](#) (uint32\_t index, const Array< Texture > &textures)
- void [setTextures](#) (uint32\_t index, const InitializerList< Texture > &textures)
- void [setSurfaceTexture](#) (uint32\_t index, Texture &texture)  
*set surfaces*
- void [setSurfaceTexture](#) (uint32\_t index, Texture &texture, const Slice &slice, Format format=Format←Unknown)
- void [setSurfaceTextures](#) (uint32\_t index, const Array< Texture > &textures)
- void [setSurfaceTextures](#) (uint32\_t index, const InitializerList< Texture > &textures)
- void \* [getUniformData](#) (uint32\_t index, size\_t size)  
*set uniforms*
- void [setUniformData](#) (uint32\_t index, const void \*src, size\_t size)



- void **setUniformBuffer** (uint32\_t index, [Buffer](#) &buffer, size\_t offset=0, size\_t size=0)
- void **setUniformOffset** (uint32\_t index, size\_t offset, bool relative=false)
- void **setUniformBuffers** (uint32\_t index, const Array< [Buffer](#) > &buffers)
- void **setUniformBuffers** (uint32\_t index, const Array< [Buffer](#) > &buffers, const Array< size\_t > &offsets)
- void **setUniformBuffers** (uint32\_t index, const InitializerList< [Buffer](#) > &buffers)
- void **setUniformBuffers** (uint32\_t index, const InitializerList< [Buffer](#) > &buffers, const InitializerList< size\_t > &offsets)
- template<class Type >  
void **setUniform** (uint32\_t index, const Array< Type > &data)
- template<class Type >  
void **setUniform** (uint32\_t index, const Type &data)
- void \* **getStorageData** (uint32\_t index, size\_t size)  
*set storages*
- void **setStorageData** (uint32\_t index, const void \*src, size\_t size)
- void **setStorageBuffer** (uint32\_t index, [Buffer](#) &buffer, size\_t offset=0, size\_t size=0)
- void **setStorageOffset** (uint32\_t index, size\_t offset, bool relative=false)
- void **setStorageBuffers** (uint32\_t index, const InitializerList< [Buffer](#) > &buffers)
- void **setStorageBuffers** (uint32\_t index, const InitializerList< [Buffer](#) > &buffers, const InitializerList< size\_t > &offsets)
- void **setStorageBuffers** (uint32\_t index, const Array< [Buffer](#) > &buffers)
- void **setStorageBuffers** (uint32\_t index, const Array< [Buffer](#) > &buffers, const Array< size\_t > &offsets)
- template<class Type >  
void **setStorage** (uint32\_t index, const Array< Type > &data)
- template<class Type >  
void **setStorage** (uint32\_t index, const Type &data)
- void **setTracing** (uint32\_t index, [Tracing](#) &tracing)  
*set tracings*
- void **setTracings** (uint32\_t index, const Array< [Tracing](#) > &tracings)
- void **setTracings** (uint32\_t index, const InitializerList< [Tracing](#) > &tracings)
- void **setTexelBuffer** (uint32\_t index, [Buffer](#) &buffer)  
*set texel buffers*
- void **setTexelBuffers** (uint32\_t index, const Array< [Buffer](#) > &buffers)
- void **setTexelBuffers** (uint32\_t index, const InitializerList< [Buffer](#) > &buffers)
- void **setTextureTable** (uint32\_t index, [TextureTable](#) &table)  
*set texture tables*
- void **setTextureTables** (uint32\_t index, const Array< [TextureTable](#) > &tables)
- void **setTextureTables** (uint32\_t index, const InitializerList< [TextureTable](#) > &tables)
- void **setStorageTable** (uint32\_t index, [BufferTable](#) &table)  
*set storage tables*
- void **setStorageTables** (uint32\_t index, const Array< [BufferTable](#) > &tables)
- void **setStorageTables** (uint32\_t index, const InitializerList< [BufferTable](#) > &tables)
- void \* **getIndirectData** (size\_t size)  
*set indirect*
- void **setIndirectData** (const void \*src, size\_t size)
- void **setIndirectBuffer** ([Buffer](#) &buffer, size\_t offset=0)
- void **setIndirectOffset** (size\_t offset, bool relative=false)
- template<class Type >  
void **setIndirect** (const Type &data)
- void **dispatch** (uint32\_t width, uint32\_t height=1, uint32\_t depth=1)  
*dispatch kernel*
- void **dispatch** (const [Texture](#) &texture)
- void **dispatch** (const [Size](#) &size)
- void **dispatchIndirect** ()  
*dispatch kernel indirect*

- bool **setBuffer** (**Buffer** &buffer, size\_t offset, const void \*src, size\_t size)
  - set buffer data*
- bool **setBuffer** (**Buffer** &buffer, const void \*src, size\_t size)
- bool **setBuffer** (**Buffer** &buffer, const void \*src)
- bool **copyBuffer** (**Buffer** &buffer, size\_t dest\_offset, **Buffer** &src, size\_t src\_offset, size\_t size)
  - copy buffer data*
- bool **copyBuffer** (**Buffer** &buffer, size\_t dest\_offset, **Buffer** &src, size\_t size)
- bool **copyBuffer** (**Buffer** &buffer, **Buffer** &src, size\_t size)
- bool **copyBuffer** (**Buffer** &buffer, **Buffer** &src)
- bool **clearBuffer** (**Buffer** &buffer, Format format, size\_t offset, const void \*src, size\_t size)
  - clear buffer data*
- bool **clearBuffer** (**Buffer** &buffer, Format format, const void \*src, size\_t size)
- bool **clearBuffer** (**Buffer** &buffer, Format format, const void \*src)
- bool **clearBuffer** (**Buffer** &buffer)
- bool **setTexture** (**Texture** &texture, const **Origin** &dest\_origin, const **Slice** &dest\_slice, const **Image** &image, const **Slice** &src\_slice)
  - set texture data*
- bool **setTexture** (**Texture** &texture, const **Origin** &dest\_origin, const **Image** &image)
- bool **setTexture** (**Texture** &texture, const **Slice** &dest\_slice, const **Image** &image)
- bool **setTexture** (**Texture** &texture, const **Image** &image)
- bool **copyTexture** (**Texture** &texture, const **Origin** &dest\_origin, const **Slice** &dest\_slice, **Texture** &src, const **Region** &src\_region, const **Slice** &src\_slice)
  - copy texture data*
- bool **copyTexture** (**Texture** &texture, const **Origin** &dest\_origin, **Texture** &src, const **Region** &src\_region)
- bool **copyTexture** (**Texture** &texture, const **Slice** &dest\_slice, **Texture** &src, const **Slice** &src\_slice)
- bool **copyTexture** (**Texture** &texture, **Texture** &src)
- bool **clearTexture** (**Texture** &texture, const **Region** &region, const **Slice** &slice, const void \*src)
  - clear texture data*
- bool **clearTexture** (**Texture** &texture, const **Region** &region, const void \*src)
- bool **clearTexture** (**Texture** &texture, const **Slice** &slice, const void \*src)
- bool **clearTexture** (**Texture** &texture, const void \*src)
- void **barrier** (**Texture** &texture)
  - resource barriers*
- void **barrier** (**Buffer** &buffer)
- void **barrier** (const Array< **Texture** > &textures)
- void **barrier** (const Array< **Buffer** > &buffers)
- void **barrier** (const InitializerList< **Texture** > &textures)
- void **barrier** (const InitializerList< **Buffer** > &buffers)
- void **beginConditional** (**Buffer** &buffer, size\_t offset)
  - begin/end conditional*
- void **endConditional** ()
- bool **beginQuery** (**Query** &query)
  - begin/end query*
- void **endQuery** (**Query** &query)

### 5.39.1 Detailed Description

**Compute** class

## 5.40 Tellusim::RadixMap&lt; Key, Type, Size &gt;::ConstIterator Class Reference

Constant iterator.

```
#include <core/TellusimRadix.h>
```

## Public Member Functions

- **ConstIterator** (const [Iterator](#) &it)
- void **clear** ()
- [ConstIterator](#) & **operator=** (const [Iterator](#) &it)
- [ConstIterator](#) & **operator=** (const [ConstIterator](#) &it)
- **operator bool** () const
- bool **operator==** (const [ConstIterator](#) &it) const
- bool **operator!=** (const [ConstIterator](#) &it) const
- [ConstIterator](#) & **operator++** ()
- [ConstIterator](#) & **operator--** ()
- [ConstIterator](#) **operator++** (int32\_t)
- [ConstIterator](#) **operator--** (int32\_t)
- [ConstIterator](#) **next** ()
- [ConstIterator](#) **prev** ()
- const Type & **operator\*** () const
- const Type \* **operator->** () const
- const Type & **get** () const

## Friends

- class **RadixMap**

## 5.40.1 Detailed Description

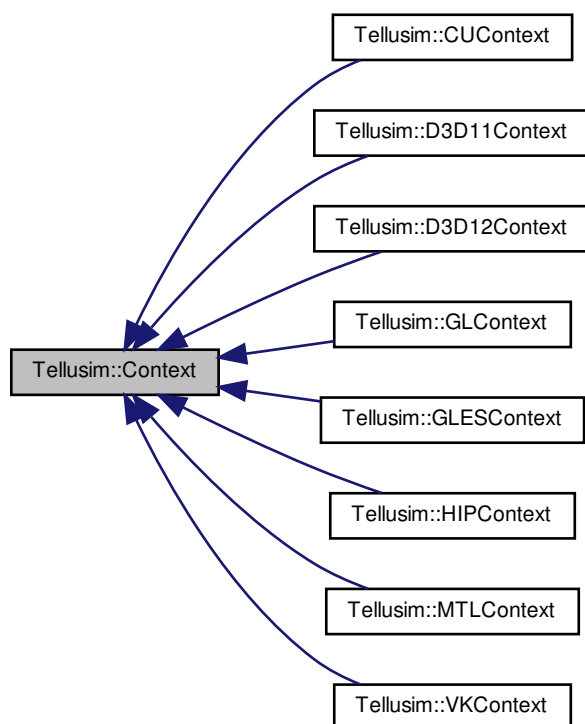
```
template<class Key, class Type, uint32_t Size = 32>
class Tellusim::RadixMap< Key, Type, Size >::ConstIterator
```

Constant iterator.

## 5.41 Tellusim::Context Class Reference

```
#include <platform/TellusimContext.h>
```

Inheritance diagram for Tellusim::Context:



#### Public Member Functions

- [Context](#) ()  
*context constructor*
- **Context** (Platform platform, uint32\_t index=Maxu32)
- Platform [getPlatform](#) () const  
*context platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*context device index*
- bool [isCreated](#) () const  
*check context*
- bool [create](#) ()  
*create context*
- bool [flush](#) ()  
*flush context*
- bool **finish** ()

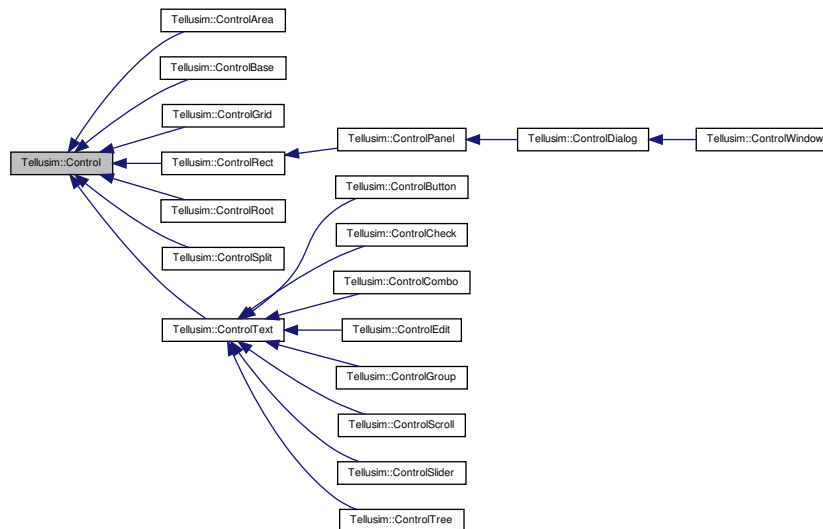
#### 5.41.1 Detailed Description

##### [Context](#)

## 5.42 Tellusim::Control Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::Control:



## Public Types

- enum [State](#) {  
**StateUnknown** = 0,  
**StateNormal**,  
**StateFocused**,  
**StatePressed**,  
**StateDisabled**,  
**NumStates** }

*Control* states.

- enum [Mesh](#) {  
**MeshCheck** = 0,  
**MeshButton**,  
**MeshSliderLine**,  
**MeshHScrollLine**,  
**MeshVScrollLine**,  
**MeshSliderHandle**,  
**MeshHScrollHandle**,  
**MeshVScrollHandle**,  
**MeshBackground**,  
**MeshSelection**,  
**MeshBorder**,  
**MeshFrame**,  
**NumMeshes** }

*Control* meshes.

- enum **Align** {  
**AlignNone** = 0,  
**AlignLeft** = (1 << 0),  
**AlignRight** = (1 << 1),  
**AlignBottom** = (1 << 2),  
**AlignTop** = (1 << 3),  
**AlignCenterX** = (1 << 4),  
**AlignCenterY** = (1 << 5),  
**AlignExpandX** = (1 << 6),  
**AlignExpandY** = (1 << 7),  
**AlignOverlap** = (1 << 8),  
**AlignSpacer** = (1 << 9),  
**AlignAspect** = (1 << 10),  
**AlignLeftBottom** = (AlignLeft | AlignBottom),  
**AlignLeftTop** = (AlignLeft | AlignTop),  
**AlignRightBottom** = (AlignRight | AlignBottom),  
**AlignRightTop** = (AlignRight | AlignTop),  
**AlignCenter** = (AlignCenterX | AlignCenterY),  
**AlignExpand** = (AlignExpandX | AlignExpandY),  
**NumAligns** = 11 }

*Control alignments.*

- enum **Button** {  
**ButtonNone** = 0,  
**ButtonLeft** = (1 << 0),  
**ButtonLeft2** = (1 << 1),  
**ButtonRight** = (1 << 2),  
**ButtonRight2** = (1 << 3),  
**ButtonMiddle** = (1 << 4),  
**ButtonMiddle2** = (1 << 5),  
**NumButtons** = 6 }

*Control buttons.*

- enum **Axis** {  
**AxisUnknown** = 0,  
**AxisX**,  
**AxisY**,  
**AxisZ**,  
**AxisW**,  
**NumAxes** }

*Control axes.*

- enum **Key** {  
**KeyNone** = 128,  
**KeyTab**,  
**KeyBackspace**,  
**KeyDelete**,  
**KeyInsert**,  
**KeyReturn**,  
**KeyPrior**,  
**KeyNext**,  
**KeyEnd**,  
**KeyHome**,  
**KeyUp**,  
**KeyDown**,  
**KeyLeft**,  
**KeyRight**,  
**KeyShift**,  
**KeyCtrl**,  
**KeyAlt**,  
**KeyCmd**,

**NumKeys,**  
**KeyOption** = KeyCtrl }  
*Control* keys.

#### Public Member Functions

- **Control** (*Control* \*parent)
- **Control** (*Control* \*parent, float32\_t width, float32\_t height=0.0f)
- Type *getType* () const  
*control type*
- const char \* **getTypeName** () const
- bool **isUnknown** () const
- bool **isRoot** () const
- bool **isText** () const
- bool **isRect** () const
- bool **isGrid** () const
- bool **isGroup** () const
- bool **isPanel** () const
- bool **isDialog** () const
- bool **isWindow** () const
- bool **isCheck** () const
- bool **isCombo** () const
- bool **isButton** () const
- bool **isSlider** () const
- bool **isScroll** () const
- bool **isSplit** () const
- bool **isArea** () const
- bool **isTree** () const
- bool **isEdit** () const
- void *setAlign* (*Align* align)  
*control alignment*
- *Align* **getAlign** () const
- bool **hasAlign** (*Align* align) const
- bool **hasAligns** (*Align* aligns) const
- bool **isSpacer** () const
- void *setEnabled* (bool enabled)  
*control enabled flag*
- bool **isEnabled** () const
- bool **wasEnabled** () const
- bool **wasUpdated** () const
- void *setDisabled* (bool disabled)  
*control disabled flag*
- bool **isDisabled** () const
- *Canvas* **getCanvas** () const  
*control canvas*
- const *ControlRoot* **getRoot** () const  
*control root*
- *ControlRoot* **getRoot** ()
- const *ControlPanel* **getPanel** () const  
*control panel*
- *ControlPanel* **getPanel** ()
- uint32\_t **setParent** (*Control* &parent)

*control parent*

- const [Control](#) **getParent** () const
- [Control](#) **getParent** ()
- bool **isParentEnabled** () const
- bool **isParentDisabled** () const
- uint32\_t **addChild** ([Control](#) &child)

*control children*

- bool **removeChild** ([Control](#) &child)
- bool **raiseChild** ([Control](#) &child)
- bool **lowerChild** ([Control](#) &child)
- void **releaseChildren** ()
- uint32\_t **findChild** (const [Control](#) &child) const
- bool **isChild** (const [Control](#) &child) const
- uint32\_t **getNumChildren** () const
- const Array< [Control](#) > **getChildren** () const
- Array< [Control](#) > **getChildren** ()
- const [Control](#) **getChild** (uint32\_t index) const
- [Control](#) **getChild** (uint32\_t index)
- void **setSize** (const [Vector2f](#) &size)

*control size*

- void **setSize** (float32\_t width, float32\_t height)
- const [Vector2f](#) & **getSize** () const
- float32\_t **getWidth** () const
- float32\_t **getHeight** () const
- void **setPosition** (const [Vector3f](#) &position)

*control position*

- void **setPosition** (float32\_t x, float32\_t y, float32\_t z=0.0f)
- const [Vector3f](#) & **getPosition** () const
- float32\_t **getPositionX** () const
- float32\_t **getPositionY** () const
- void **setOffset** (const [Vector3f](#) &offset)

*control offset*

- void **setOffset** (float32\_t x, float32\_t y, float32\_t z=0.0f)
- const [Vector3f](#) & **getOffset** () const
- float32\_t **getOffsetX** () const
- float32\_t **getOffsetY** () const
- void **setMargin** (float32\_t value)

*control margin*

- void **setMargin** (float32\_t horizontal, float32\_t vertical)
- void **setMargin** (float32\_t left, float32\_t right, float32\_t bottom, float32\_t top)
- void **setMargin** (const [Rect](#) &margin)
- const [Rect](#) & **getMargin** () const
- const [Rect](#) & **getRect** () const

*control rectangle*

- [State](#) **getState** () const

*control state***Static Public Member Functions**

- static const char \* **getTypeName** (Type type)



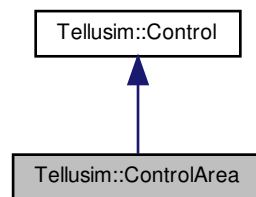
## 5.42.1 Detailed Description

[Control](#) base class

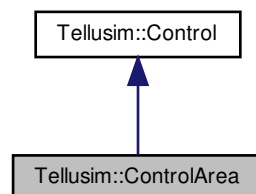
## 5.43 Tellusim::ControlArea Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlArea:



Collaboration diagram for Tellusim::ControlArea:



## Public Member Functions

- **ControlArea** ([Control](#) \*parent)
- **ControlArea** ([Control](#) \*parent, uint32\_t columns)
- **ControlArea** ([Control](#) \*parent, bool horizontal, bool vertical)
- **ControlArea** ([Control](#) \*parent, uint32\_t columns, float32\_t x, float32\_t y)
- void [setAbsolute](#) (bool absolute)  
*absolute flag*
- bool **isAbsolute** () const
- void [setScalable](#) (bool scalable)  
*scalable flag*

- bool **isScalable** () const
- void **setScrollable** (bool scrollable)
  - scrollable flag*
- bool **isScrollable** () const
- void **setScale** (float32\_t scale)
  - area scale*
- float32\_t **getScale** () const
- void **setScaleRange** (float32\_t min, float32\_t max)
  - scale range*
- float32\_t **getMinScale** () const
- float32\_t **getMaxScale** () const
- void **setHorizontalStep** (float64\_t step)
  - area step*
- void **setVerticalStep** (float64\_t step)
- void **setStep** (float64\_t horizontal, float64\_t vertical)
- float64\_t **getHorizontalStep** () const
- float64\_t **getVerticalStep** () const
- void **setHorizontalValue** (float64\_t value)
  - area value*
- void **setVerticalValue** (float64\_t value)
- void **setValue** (float64\_t horizontal, float64\_t vertical)
- float64\_t **getHorizontalValue** () const
- float64\_t **getVerticalValue** () const
- void **setFrameAlign** (Align align)
  - frame alignment*
- **Align** **getFrameAlign** () const
- float64\_t **getHorizontalFrame** () const
  - area frame*
- float64\_t **getVerticalFrame** () const
- float64\_t **getHorizontalRange** () const
  - area range*
- float64\_t **getVerticalRange** () const
- void **setHorizontalEnabled** (bool enabled, bool dynamic=false)
  - horizontal scroll*
- bool **isHorizontalEnabled** () const
- bool **isHorizontalDynamic** () const
- bool **isHorizontalHidden** () const
- const **ControlScroll** **getHorizontalScroll** () const
- **ControlScroll** **getHorizontalScroll** ()
- void **setVerticalEnabled** (bool enabled, bool dynamic=false)
  - vertical scroll*
- bool **isVerticalEnabled** () const
- bool **isVerticalDynamic** () const
- bool **isVerticalHidden** () const
- const **ControlScroll** **getVerticalScroll** () const
- **ControlScroll** **getVerticalScroll** ()
- bool **setFontSize** (uint32\_t size)
  - font style*
- uint32\_t **getFontSize** () const
- bool **setFontStyle** (const **FontStyle** &style)
- const **FontStyle** & **getFontStyleConst** () const
- const **FontStyle** & **getFontStyle** () const
- **FontStyle** & **getFontStyle** ()

- void **setColumns** (uint32\_t columns)  
*number of columns*
- uint32\_t **getColumns** () const
- void **setSpacing** (const Vector2f &spacing)  
*grid spacing*
- void **setSpacing** (float32\_t x, float32\_t y)
- const Vector2f & **getSpacing** () const
- void **setColumnRatio** (uint32\_t index, float32\_t ratio)  
*grid column ratio*
- float32\_t **getColumnRatio** (uint32\_t index) const
- const Vector2f & **getControlsSize** () const  
*controls size*
- const Vector2f & **getControlsOffset** () const  
*controls offset*
- const Rect & **getViewRect** () const  
*view rectangle*

#### Additional Inherited Members

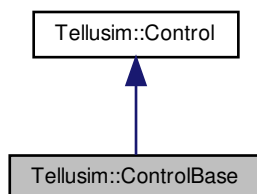
##### 5.43.1 Detailed Description

[ControlArea](#) class

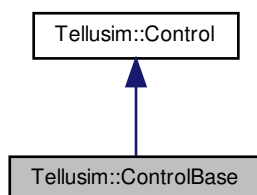
## 5.44 Tellusim::ControlBase Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlBase:



Collaboration diagram for Tellusim::ControlBase:



### Protected Member Functions

- **ControlBase** ([Control](#) \*parent)
- virtual void [clear](#) ()  
*clear control*
- void [create](#) ()  
*create control*
- [CanvasText](#) [create\\_text](#) ()  
*create canvas*
- [CanvasMesh](#) [create\\_mesh](#) ()
- virtual bool [is\\_batch](#) () const  
*control batch*
- bool [get\\_disabled](#) ([ControlRoot](#) &root) const  
*disabled control*
- [Vector3f](#) [get\\_position](#) (const [Rect](#) &region, uint32\_t scale) const  
*control position*
- virtual void [update\\_enabled](#) (bool enabled)  
*update control*
- virtual void [update\\_disabled](#) (bool disabled)
- virtual void [update\\_style](#) (const [FontStyle](#) &style)
- virtual void [update\\_expand](#) ([ControlRoot](#) &root, const [Rect](#) &region)
- virtual void [update\\_position](#) ([ControlRoot](#) &root, const [Vector2f](#) &offset)
- virtual void [update\\_mouse](#) ([ControlRoot](#) &root, [Axis](#) axis, float32\_t delta)
- virtual bool [update\\_keyboard](#) ([ControlRoot](#) &root, uint32\_t key, uint32\_t code)
- virtual void [update\\_rectangle](#) ([ControlRoot](#) &root, int32\_t &order, uint32\_t scale)
- virtual bool [update](#) ([ControlRoot](#) &root, const [Rect](#) &region, const [Rect](#) &view, uint32\_t scale)
- void [update\\_text](#) ([ControlRoot](#) &root, [CanvasText](#) &canvas\_text, const [Color](#) &color, [State](#) state, const [Vector3f](#) &position) const  
*update canvas*
- void [update\\_mesh](#) ([ControlRoot](#) &root, [CanvasMesh](#) &canvas\_mesh, [Mesh](#) mesh, [State](#) state, const [Rect](#) &rect, uint32\_t scale, bool clear=true) const
- [State](#) [set\\_state](#) ([ControlRoot](#) &root, [State](#) state)  
*set control state*
- void [set\\_rect](#) (const [Rect](#) &rect)  
*set control rect*

## Additional Inherited Members

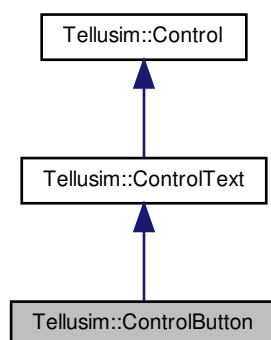
## 5.44.1 Detailed Description

[ControlBase](#) class

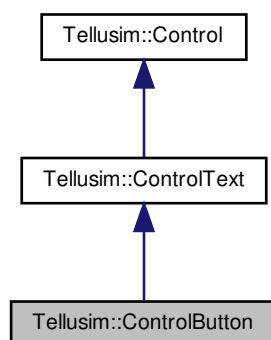
## 5.45 Tellusim::ControlButton Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlButton:



Collaboration diagram for Tellusim::ControlButton:



## Public Types

- using [PressedCallback](#) = Function< void([ControlButton](#), float32\_t x, float32\_t y)>  
*pressed callback*
- using [ReleasedCallback](#) = Function< void([ControlButton](#), float32\_t x, float32\_t y)>  
*released callback*
- using [ClickedCallback](#) = Function< void([ControlButton](#))>  
*clicked callback*

## Public Member Functions

- **ControlButton** ([Control](#) \*parent)
- **ControlButton** ([Control](#) \*parent, const char \*text)
- **ControlButton** ([Control](#) \*parent, const [String](#) &text)
- void **setBackground** (bool background)  
*background flag*
- bool **getBackground** () const
- void **setButtonMode** ([CanvasElement::Mode](#) mode)  
*control mode*
- [CanvasElement::Mode](#) **getButtonMode** () const
- void **setButtonRadius** (float32\_t radius)  
*button radius*
- float32\_t **getButtonRadius** () const
- void **setButtonColor** (const [Color](#) &color)  
*button color*
- const [Color](#) & **getButtonColor** () const
- void **setStrokeStyle** (const [StrokeStyle](#) &style)  
*stroke style*
- const [StrokeStyle](#) & **getStrokeStyleConst** () const
- const [StrokeStyle](#) & **getStrokeStyle** () const
- [StrokeStyle](#) & **getStrokeStyle** ()
- void **setGradientStyle** (const [GradientStyle](#) &style)  
*gradient style*
- const [GradientStyle](#) & **getGradientStyleConst** () const
- const [GradientStyle](#) & **getGradientStyle** () const
- [GradientStyle](#) & **getGradientStyle** ()
- void **setPressedCallback** (const [PressedCallback](#) &func)
- [PressedCallback](#) **getPressedCallback** () const
- bool **isPressed** ()
- void **setReleasedCallback** (const [ReleasedCallback](#) &func)
- [ReleasedCallback](#) **getReleasedCallback** () const
- bool **isReleased** ()
- void **setClickedCallback** (const [ClickedCallback](#) &func)
- [ClickedCallback](#) **getClickedCallback** () const
- bool **isClicked** ()
- [CanvasRect](#) **getCanvasRect** ()  
*canvas elements*
- [CanvasMesh](#) **getCanvasMesh** ()

## Additional Inherited Members

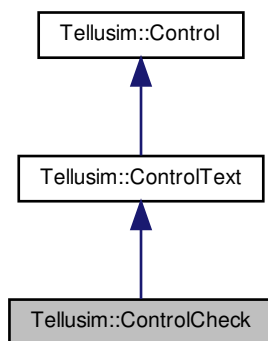
## 5.45.1 Detailed Description

[ControlButton](#) class

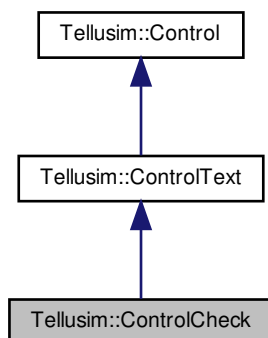
## 5.46 Tellusim::ControlCheck Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlCheck:



Collaboration diagram for Tellusim::ControlCheck:



## Public Types

- using [ClickedCallback](#) = Function< void([ControlCheck](#))>  
*clicked callback*

## Public Member Functions

- **ControlCheck** ([Control](#) \*parent)
- **ControlCheck** ([Control](#) \*parent, const char \*text)
- **ControlCheck** ([Control](#) \*parent, const [String](#) &text)
- **ControlCheck** ([Control](#) \*parent, const char \*text, bool checked)
- **ControlCheck** ([Control](#) \*parent, const [String](#) &text, bool checked)
- void [setCheckText](#) (const char \*text)  
*check text*
- void **setCheckText** (const [String](#) &text)
- [String](#) [getCheckText](#) () const
- void [setCheckColor](#) (const [Color](#) &color)  
*check color*
- const [Color](#) & **getCheckColor** () const
- void [setCheckedColor](#) (const [Color](#) &color)  
*checked color*
- const [Color](#) & **getCheckedColor** () const
- bool [switchChecked](#) (bool callback=false)  
*checked state*
- void **setChecked** (bool checked, bool callback=false)
- bool **isChecked** () const
- void **setClickedCallback** (const [ClickedCallback](#) &func)
- [ClickedCallback](#) **getClickedCallback** () const
- bool **isClicked** ()
- [CanvasMesh](#) [getCanvasMesh](#) ()  
*canvas elements*

## Additional Inherited Members

## 5.46.1 Detailed Description

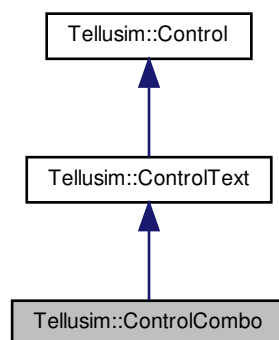
[ControlCheck](#) class

## 5.47 Tellusim::ControlCombo Class Reference

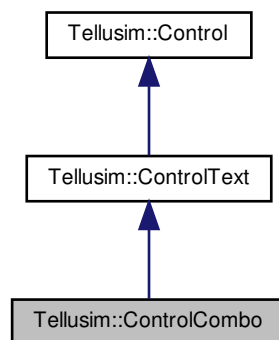
```
#include <interface/TellusimControls.h>
```



Inheritance diagram for Tellusim::ControlCombo:



Collaboration diagram for Tellusim::ControlCombo:



### Public Types

- using [ClickedCallback](#) = Function< void([ControlCombo](#))>  
*clicked callback*
- using [ChangedCallback](#) = Function< void([ControlCombo](#))>  
*changed callback*

### Public Member Functions

- **ControlCombo** ([Control](#) \*parent)
- **ControlCombo** ([Control](#) \*parent, const InitializerList< const char \*> &items)
- **ControlCombo** ([Control](#) \*parent, const InitializerList< const char \*> &items, uint32\_t index)

- void **setMultiSelection** (bool multi\_selection)  
*multi-selection flag*
- bool **isMultiSelection** () const
- void **setComboText** (const char \*text)  
*combo text*
- void **setComboText** (const **String** &text)
- **String** **getComboText** () const
- void **setComboColor** (const **Color** &color)  
*combo color*
- const **Color** & **getComboColor** () const
- void **setItemsSpacing** (float32\_t spacing)  
*items spacing*
- float32\_t **getItemsSpacing** () const
- void **clearItems** ()  
*combo items*
- uint32\_t **addItem** (const char \*text)
- uint32\_t **addItem** (const **String** &text)
- void **addItem** (uint32\_t index, const char \*text)
- void **addItem** (uint32\_t index, const **String** &text)
- void **addItems** (const InitializerList< const char \*> &items)
- void **removeItem** (uint32\_t index)
- uint32\_t **getNumItems** () const
- bool **switchItemSelected** (uint32\_t index)  
*item selected flag*
- void **setItemSelected** (uint32\_t index, bool selected)
- bool **isItemSelected** (uint32\_t index) const
- void **setItemText** (uint32\_t index, const char \*text)  
*item text*
- void **setItemText** (uint32\_t index, const **String** &text)
- **String** **getItemText** (uint32\_t index) const
- uint32\_t **findItemText** (const char \*text) const
- uint32\_t **findItemText** (const **String** &text) const
- void **setItemColor** (uint32\_t index, const **Color** &color)  
*item color*
- const **Color** & **getItemColor** (uint32\_t index) const
- void **setCurrentIndex** (uint32\_t index, bool callback=false)  
*current item*
- void **setCurrentText** (const char \*text, bool callback=false)
- void **setCurrentText** (const **String** &text, bool callback=false)
- uint32\_t **getCurrentIndex** () const
- **String** **getCurrentText** () const
- void **setClickedCallback** (const **ClickedCallback** &func)
- **ClickedCallback** **getClickedCallback** () const
- bool **isClicked** ()
- void **setChangedCallback** (const **ChangedCallback** &func)
- **ChangedCallback** **getChangedCallback** () const
- bool **isChanged** ()
- **CanvasMesh** **getCanvasMesh** ()  
*canvas elements*

## Additional Inherited Members

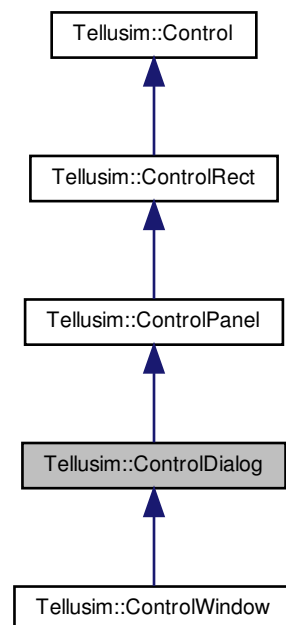
## 5.47.1 Detailed Description

[ControlCombo](#) class

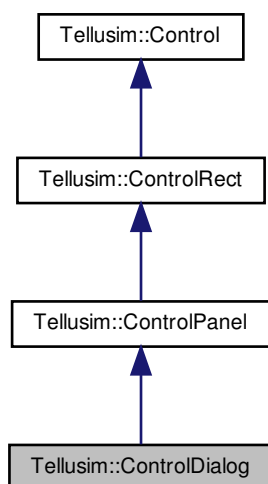
## 5.48 Tellusim::ControlDialog Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlDialog:



Collaboration diagram for Tellusim::ControlDialog:



### Public Types

- using [UpdatedCallback](#) = Function< void([ControlDialog](#))>  
*updated callback*

### Public Member Functions

- **ControlDialog** ([Control](#) \*parent)
- **ControlDialog** ([Control](#) \*parent, uint32\_t columns)
- **ControlDialog** ([Control](#) \*parent, uint32\_t columns, float32\_t x, float32\_t y)
- void [setConstrained](#) (bool constrained)  
*constrained flag*
- bool **isConstrained** () const
- void [setResizable](#) (bool resizable)  
*resizable flag*
- bool **isResizable** () const
- void [setResizeArea](#) (float32\_t area)  
*resize area*
- float32\_t **getResizeArea** () const
- [Align](#) [getResizeAlign](#) () const  
*resize alignment*
- bool **hasResizeAlign** ([Align](#) align) const
- bool **hasResizeAligns** ([Align](#) aligns) const
- void [setMousePosition](#) (const [Vector2f](#) &position)  
*mouse position*
- const [Vector2f](#) & **getMousePosition** () const
- void **setUpdatedCallback** (const [UpdatedCallback](#) &func)
- [UpdatedCallback](#) **getUpdatedCallback** () const
- bool **isUpdated** ()

## Additional Inherited Members

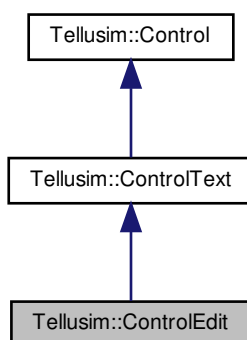
## 5.48.1 Detailed Description

[ControlDialog](#) class

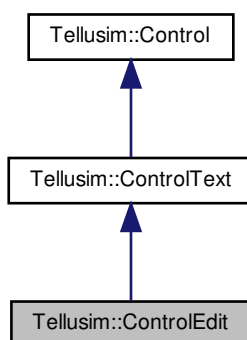
## 5.49 Tellusim::ControlEdit Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlEdit:



Collaboration diagram for Tellusim::ControlEdit:



## Public Types

- enum `EditMode` {  
**EditModeText**,  
**EditModePassword**,  
**EditModeNumber**,  
**EditModeSigned**,  
**EditModeUnsigned**,  
**EditModeHexadecimal** }  
*edit mode*
- using `ClickedCallback` = Function< void(`ControlEdit`)>  
*clicked callback*
- using `ChangedCallback` = Function< void(`ControlEdit`)>  
*changed callback*
- using `ReturnedCallback` = Function< void(`ControlEdit`)>  
*returned callback*

## Public Member Functions

- **ControlEdit** (`Control` \*parent)
- **ControlEdit** (`Control` \*parent, const char \*text)
- **ControlEdit** (`Control` \*parent, const `String` &text)
- void **setFrame** (bool frame)  
*frame flag*
- bool **getFrame** () const
- void **setBackground** (bool background)  
*background flag*
- bool **getBackground** () const
- void **setEditColor** (const `Color` &color)  
*edit color*
- const `Color` & **getEditColor** () const
- void **setEditMode** (`EditMode` mode)
- `EditMode` **getEditMode** () const
- void **setPasswordCode** (uint32\_t code)  
*password code*
- uint32\_t **getPasswordCode** () const
- uint32\_t **getNumCodes** () const  
*edit codes*
- const uint32\_t \* **getCodes** () const
- void **setCurrentIndex** (uint32\_t index, uint32\_t selection\_index=Maxu32)  
*current index*
- uint32\_t **getCurrentIndex** () const
- uint32\_t **getSelectionIndex** () const
- void **setSelection** (bool current=false)  
*selected text*
- void **clearSelection** ()
- `String` **getSelectedText** () const
- void **setClickedCallback** (const `ClickedCallback` &func)
- `ClickedCallback` **getClickedCallback** () const
- bool **isClicked** ()
- void **setChangedCallback** (const `ChangedCallback` &func)
- `ChangedCallback` **getChangedCallback** () const
- bool **isChanged** ()

- void **setReturnedCallback** (const [ReturnedCallback](#) &func)
- [ReturnedCallback](#) **getReturnedCallback** () const
- bool **isReturned** ()
- [CanvasMesh](#) **getCanvasMesh** ()  
*canvas elements*

#### Additional Inherited Members

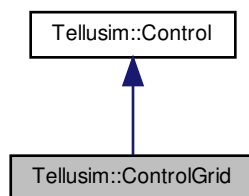
##### 5.49.1 Detailed Description

[ControlEdit](#) class

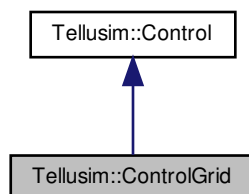
## 5.50 Tellusim::ControlGrid Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlGrid:



Collaboration diagram for Tellusim::ControlGrid:



## Public Member Functions

- **ControlGrid** ([Control](#) \*parent)
- **ControlGrid** ([Control](#) \*parent, uint32\_t columns)
- **ControlGrid** ([Control](#) \*parent, uint32\_t columns, float32\_t x, float32\_t y=0.0f)
- void **setColumns** (uint32\_t columns)  
*number of columns*
- uint32\_t **getColumns** () const
- void **setSpacing** (const [Vector2f](#) &spacing)  
*grid spacing*
- void **setSpacing** (float32\_t x, float32\_t y)
- const [Vector2f](#) & **getSpacing** () const
- void **setColumnRatio** (uint32\_t index, float32\_t ratio)  
*grid column ratio*
- float32\_t **getColumnRatio** (uint32\_t index) const
- const [Vector2f](#) & **getControlsSize** () const  
*controls size*

## Additional Inherited Members

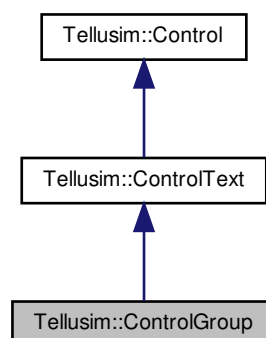
## 5.50.1 Detailed Description

[ControlGrid](#) class

## 5.51 Tellusim::ControlGroup Class Reference

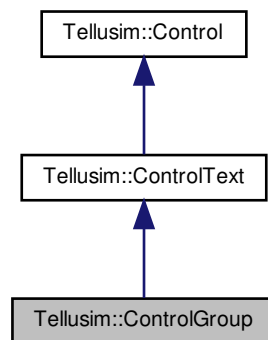
```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlGroup:





Collaboration diagram for Tellusim::ControlGroup:



#### Public Types

- using [ClickedCallback](#) = Function< void([ControlGroup](#))>  
*clicked callback*

#### Public Member Functions

- **ControlGroup** ([Control](#) \*parent, bool above=false)
- **ControlGroup** ([Control](#) \*parent, const char \*text, bool above=false)
- **ControlGroup** ([Control](#) \*parent, const [String](#) &text, bool above=false)
- **ControlGroup** ([Control](#) \*parent, const char \*text, uint32\_t columns, bool above=false)
- **ControlGroup** ([Control](#) \*parent, const char \*text, uint32\_t columns, float32\_t x, float32\_t y, bool above=false)
- void [setAbove](#) (bool above, bool text=true)  
*above flag*
- bool **isAbove** () const
- bool **isBelow** () const
- void [setFoldable](#) (bool foldable)  
*foldable flag*
- bool **isFoldable** () const
- void [setExpanded](#) (bool expanded)  
*expanded flag*
- bool **isExpanded** () const
- void [setBackground](#) (bool background)  
*background flag*
- bool **getBackground** () const
- void [setGroupRadius](#) (float32\_t radius)  
*group radius*
- float32\_t **getGroupRadius** () const
- void [setGroupColor](#) (const [Color](#) &color)  
*group color*
- const [Color](#) & **getGroupColor** () const
- void [setStrokeStyle](#) (const [StrokeStyle](#) &style)

*stroke style*

- const [StrokeStyle](#) & **getStrokeStyleConst** () const
- const [StrokeStyle](#) & **getStrokeStyle** () const
- [StrokeStyle](#) & **getStrokeStyle** ()
- void **setGradientStyle** (const [GradientStyle](#) &style)

*gradient style*

- const [GradientStyle](#) & **getGradientStyleConst** () const
- const [GradientStyle](#) & **getGradientStyle** () const
- [GradientStyle](#) & **getGradientStyle** ()
- void **setFoldedText** (const char \*text)

*folded text*

- void **setFoldedText** (const [String](#) &text)
- [String](#) **getFoldedText** () const
- void **setExpandedText** (const char \*text)

*expanded text*

- void **setExpandedText** (const [String](#) &text)
- [String](#) **getExpandedText** () const
- void **setColumns** (uint32\_t columns)

*number of columns*

- uint32\_t **getColumns** () const
- void **setSpacing** (const [Vector2f](#) &spacing)

*grid spacing*

- void **setSpacing** (float32\_t x, float32\_t y)
- const [Vector2f](#) & **getSpacing** () const
- void **setColumnRatio** (uint32\_t index, float32\_t ratio)

*grid column ratio*

- float32\_t **getColumnRatio** (uint32\_t index) const
- const [Vector2f](#) & **getControlsSize** () const

*controls size*

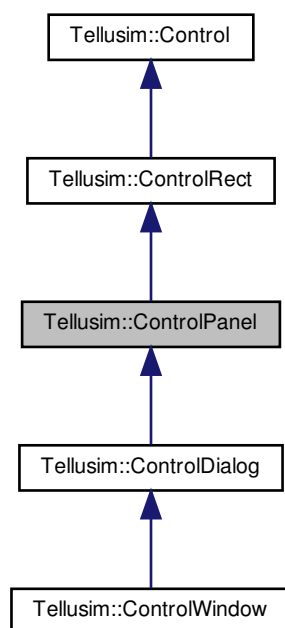
- void **setClickedCallback** (const [ClickedCallback](#) &func)
- [ClickedCallback](#) **getClickedCallback** () const
- bool **isClicked** ()
- [CanvasRect](#) **getCanvasRect** ()

*canvas elements***Additional Inherited Members****5.51.1 Detailed Description**[ControlGroup](#) class

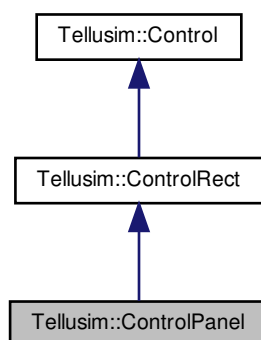
## 5.52 Tellusim::ControlPanel Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlPanel:



Collaboration diagram for Tellusim::ControlPanel:



## Public Member Functions

- **ControlPanel** ([Control](#) \*parent)
- **ControlPanel** ([Control](#) \*parent, uint32\_t columns)
- **ControlPanel** ([Control](#) \*parent, uint32\_t columns, float32\_t x, float32\_t y)
- void [setColumns](#) (uint32\_t columns)  
*number of columns*
- uint32\_t [getColumns](#) () const
- void [setSpacing](#) (const [Vector2f](#) &spacing)  
*grid spacing*
- void [setSpacing](#) (float32\_t x, float32\_t y)
- const [Vector2f](#) & [getSpacing](#) () const
- void [setColumnRatio](#) (uint32\_t index, float32\_t ratio)  
*grid column ratio*
- float32\_t [getColumnRatio](#) (uint32\_t index) const
- const [Vector2f](#) & [getControlsSize](#) () const  
*controls size*

## Additional Inherited Members

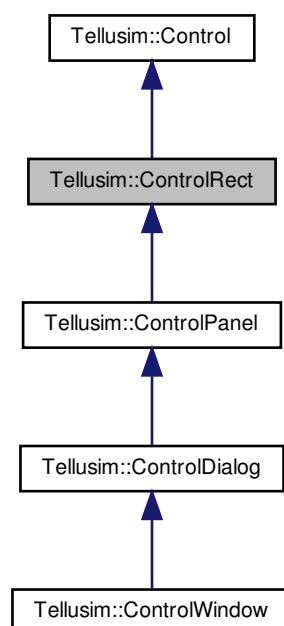
## 5.52.1 Detailed Description

[ControlPanel](#) class

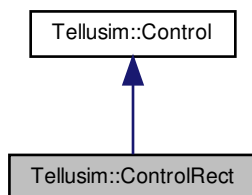
## 5.53 Tellusim::ControlRect Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlRect:



Collaboration diagram for Tellusim::ControlRect:



### Public Types

- using `PressedCallback` = `Function< void(ControlRect, float32_t x, float32_t y)>`  
*pressed callback*
- using `ReleasedCallback` = `Function< void(ControlRect, float32_t x, float32_t y)>`  
*released callback*
- using `ClickedCallback` = `Function< void(ControlRect)>`  
*clicked callback*

### Public Member Functions

- **ControlRect** (`Control` \*parent)
- **ControlRect** (`Control` \*parent, `Texture` &texture)
- **ControlRect** (`Control` \*parent, const char \*name)
- **ControlRect** (`Control` \*parent, `CanvasElement::Mode` mode)
- void **setCallback** (bool callback)  
*callback flag*
- bool **getCallback** () const
- void **setFullscreen** (bool fullscreen)  
*fullscreen flag*
- bool **isFullscreen** () const
- void **setMode** (`CanvasElement::Mode` mode)  
*control mode*
- `CanvasElement::Mode` **getMode** () const
- void **setPipeline** (`Pipeline` &pipeline)  
*control pipeline*
- void **setPipeline** (`Pipeline` &pipeline, const `CanvasElement::DrawCallback` &func)
- `Pipeline` **getPipeline** () const
- void **setRadius** (float32\_t radius)  
*control radius*
- float32\_t **getRadius** () const
- void **setColor** (const `Color` &color)  
*control color*
- void **setColor** (float32\_t r, float32\_t g, float32\_t b, float32\_t a)
- const `Color` & **getColor** () const
- void **setStrokeStyle** (const `StrokeStyle` &style)

- stroke style*
  - const [StrokeStyle](#) & **getStrokeStyleConst** () const
  - const [StrokeStyle](#) & **getStrokeStyle** () const
  - [StrokeStyle](#) & **getStrokeStyle** ()
  - void **setGradientStyle** (const [GradientStyle](#) &style)
- gradient style*
  - const [GradientStyle](#) & **getGradientStyleConst** () const
  - const [GradientStyle](#) & **getGradientStyle** () const
  - [GradientStyle](#) & **getGradientStyle** ()
  - void **setMipmap** (float32\_t mipmap)
- control mipmap*
  - float32\_t **getMipmap** () const
  - void **setFilter** ([Sampler::Filter](#) filter)
- filter mode*
  - [Sampler::Filter](#) **getFilter** () const
  - void **setAnisotropy** (uint32\_t anisotropy)
  - uint32\_t **getAnisotropy** () const
  - void **setWrapMode** ([Sampler::WrapMode](#) mode)
- wrapping mode*
  - [Sampler::WrapMode](#) **getWrapMode** () const
  - void **setBlend** ([Pipeline::BlendOp](#) op, [Pipeline::BlendFunc](#) src, [Pipeline::BlendFunc](#) dest)
- blending parameters*
  - [Pipeline::BlendOp](#) **getBlendOp** () const
  - [Pipeline::BlendFunc](#) **getBlendSrcFunc** () const
  - [Pipeline::BlendFunc](#) **getBlendDestFunc** () const
  - void **setTexture** ([Texture](#) &texture, bool linear=false)
- texture pointer*
  - [Texture](#) **getTexture** () const
  - bool **getTextureLinear** () const
  - void **setTextureName** (const char \*name)
- texture name*
  - void **setTextureName** (const [String](#) &name)
  - [String](#) **getTextureName** () const
  - void **setTextureScale** (float32\_t scale\_x, float32\_t scale\_y)
- texture scale*
  - float32\_t **getTextureScaleX** () const
  - float32\_t **getTextureScaleY** () const
  - void **setTextureFlip** (bool flip\_x, bool flip\_y)
- texture orientation*
  - bool **getTextureFlipX** () const
  - bool **getTextureFlipY** () const
  - void **setTextureProj** (bool projection)
- texture projection*
  - bool **getTextureProj** () const
  - void **setTexCoord** (const [Rect](#) &texcoord)
- texture coordinates*
  - void **setTexCoord** (float32\_t left, float32\_t right, float32\_t bottom, float32\_t top)
  - const [Rect](#) & **getTexCoord** () const
  - void **setPressedCallback** (const [PressedCallback](#) &func)
  - [PressedCallback](#) **getPressedCallback** () const
  - void **setReleasedCallback** (const [ReleasedCallback](#) &func)
  - [ReleasedCallback](#) **getReleasedCallback** () const
  - void **setClickedCallback** (const [ClickedCallback](#) &func)

- void **setClicked2Callback** (const [ClickedCallback](#) &func)
- void **setClickedRightCallback** (const [ClickedCallback](#) &func)
- [ClickedCallback](#) **getClickedCallback** () const
- [ClickedCallback](#) **getClicked2Callback** () const
- [ClickedCallback](#) **getClickedRightCallback** () const
- [CanvasRect](#) **getCanvasRect** ()  
*canvas elements*
- [CanvasMesh](#) **getCanvasMesh** ()

#### Additional Inherited Members

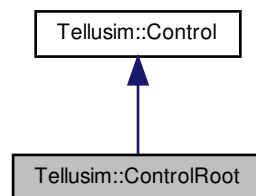
##### 5.53.1 Detailed Description

[ControlRect](#) class

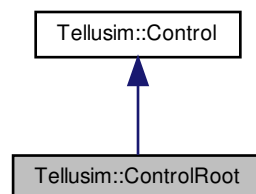
## 5.54 Tellusim::ControlRoot Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlRoot:



Collaboration diagram for Tellusim::ControlRoot:



## Public Types

- using **CopyCallback** = Function< void(**ControlRoot**, const char \*text)>  
*copy callback*
- using **PasteCallback** = Function< **String**(**ControlRoot**)>  
*paste callback*

## Public Member Functions

- **ControlRoot** (**Canvas** &canvas, bool blob=false)
- void **setViewport** (const **Viewport** &viewport)  
*root viewport*
- void **setViewport** (uint32\_t width, uint32\_t height)
- void **setViewport** (float32\_t width, float32\_t height)
- const **Viewport** & **getViewport** () const
- **String** **getFontName** () const  
*font name*
- void **setFontName** (const char \*name)
- void **setFontName** (const **String** &name)
- bool **setFontBlob** (const uint8\_t(\*blob)[256], const char \*name=nullptr)
- bool **setFontSize** (uint32\_t size, bool update=false)  
*font style*
- uint32\_t **getFontSize** () const
- bool **setFontScale** (uint32\_t scale, bool update=false)
- uint32\_t **getFontScale** () const
- bool **setFontStyle** (const **FontStyle** &style, bool update=false)
- const **FontStyle** & **getFontStyle** () const
- **FontStyle** & **getFontStyle** ()
- **String** **getTextureName** () const  
*texture parameters*
- bool **setTextureName** (const char \*name, uint32\_t width=0, uint32\_t height=0, float32\_t border=0.0f)
- bool **setTextureName** (const **String** &name, uint32\_t width=0, uint32\_t height=0, float32\_t border=0.0f)
- bool **setTextureBlob** (const uint8\_t(\*blob)[256], const char \*name=nullptr, uint32\_t width=0, uint32\_t height=0, float32\_t border=0.0f)
- float32\_t **getTextureWidth** () const
- float32\_t **getTextureHeight** () const
- void **setTextColor** (Type type, **State** state, const **Color** &color)  
*text parameters*
- void **setTextOffset** (Type type, **State** state, const **Vector3f** &offset)
- const **Color** & **getTextColor** (Type type, **State** state) const
- const **Vector3f** & **getTextOffset** (Type type, **State** state) const
- void **setMeshColor** (**Mesh** mesh, **State** state, const **Color** &color)  
*mesh parameters*
- void **setMeshRegion** (**Mesh** mesh, const **Rect** &grid, const **Rect** &region, const **Vector2f** &border)
- void **setMeshRegions** (**Mesh** mesh, const **Rect** &grid, const **Rect** &regions, const **Vector2f** &border)
- uint32\_t **getMeshColor** (**Mesh** mesh, **State** state) const
- const **Rect** & **getMeshGrid** (**Mesh** mesh) const
- const **Rect** & **getMeshMargin** (**Mesh** mesh) const
- const **Rect** & **getMeshRegion** (**Mesh** mesh, **State** state) const
- void **setGroupRadius** (float32\_t radius)  
*panel parameters*
- void **setGroupColor** (const **Color** &color)
- float32\_t **getGroupRadius** () const



- const [Color](#) & **getGroupColor** () const
- void **setPanelRadius** (float32\_t radius)
  - panel parameters*
- void **setPanelColor** (const [Color](#) &color)
- float32\_t **getPanelRadius** () const
- const [Color](#) & **getPanelColor** () const
- void **setCheckedColor** (const [Color](#) &color)
  - check parameters*
- const [Color](#) & **getCheckedColor** () const
- void **setMouse** (int32\_t x, int32\_t y, [Button](#) buttons)
  - mouse button*
- void **setMouse** (float32\_t x, float32\_t y, [Button](#) buttons)
- const [Vector2f](#) & **getMouse** () const
- [Button](#) **getMouseButtons** () const
- float32\_t **getMouseX** () const
- float32\_t **getMouseY** () const
- void **setMouseAxis** ([Axis](#) axis, float32\_t delta)
  - mouse axes*
- void **setMouseAlign** ([Align](#) align, bool clear=true)
  - mouse alignment*
- [Align](#) **getMouseAlign** () const
- bool **hasMouseAlign** ([Align](#) align) const
- bool **hasMouseAligns** ([Align](#) aligns) const
- bool **setKeyboardKey** (uint32\_t key, uint32\_t code, bool value)
  - keyboard keys*
- bool **getKeyboardKey** (uint32\_t key, bool clear=false)
- void **clearCurrentControl** ()
  - current control*
- void **setCurrentControl** ([Control](#) control, bool grab=false)
- [Control](#) **getCurrentControl** () const
- bool **getControlGrab** () const
- void **clearFocusedControl** ()
  - focused control*
- void **setFocusedControl** ([Control](#) control)
- [Control](#) **getFocusedControl** () const
- bool **isFocusedControl** () const
- void **clearMouseControl** ()
  - mouse control*
- void **setMouseControl** ([Control](#) control)
- [Control](#) **getMouseControl** () const
- bool **isMouseControl** () const
- void **clearInputControl** ()
  - input control*
- void **setInputControl** ([Control](#) control)
- [Control](#) **getInputControl** () const
- bool **isInputControl** () const
- void **clearModalControl** ()
  - modal control*
- void **setModalControl** ([Control](#) control)
- [Control](#) **getModalControl** () const
- bool **isModalControl** () const
- bool **update** (uint32\_t scale=0, int32\_t order=0)
  - update controls*

- void `setCopyText` (const char \*text)  
*copy/paste buffer*
- void `setCopyText` (const `String` &text)
- `String` `getPasteText` ()
- void `setCopyCallback` (const `CopyCallback` &func)
- `CopyCallback` `getCopyCallback` () const
- void `setPasteCallback` (const `PasteCallback` &func)
- `PasteCallback` `getPasteCallback` () const

#### Additional Inherited Members

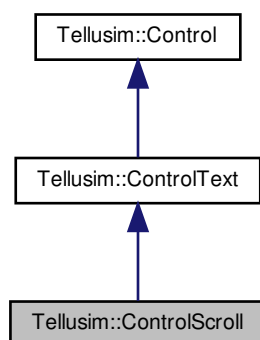
#### 5.54.1 Detailed Description

`ControlRoot` class

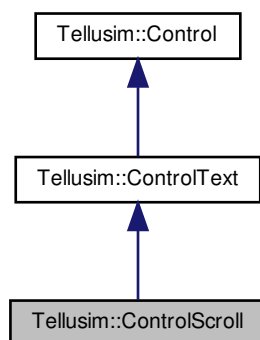
### 5.55 Tellusim::ControlScroll Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlScroll:



Collaboration diagram for Tellusim::ControlScroll:



#### Public Types

- using [ClickedCallback](#) = Function< void([ControlScroll](#))>  
*clicked callback*
- using [ChangedCallback](#) = Function< void([ControlScroll](#))>  
*changed callback*

#### Public Member Functions

- **ControlScroll** ([Control](#) \*parent, bool vertical=false)
- **ControlScroll** ([Control](#) \*parent, float64\_t value, bool vertical=false)
- **ControlScroll** ([Control](#) \*parent, float64\_t value, float64\_t frame, float64\_t range, bool vertical=false)
- **ControlScroll** ([Control](#) \*parent, uint32\_t value, uint32\_t frame, uint32\_t range, bool vertical=false)
- **ControlScroll** ([Control](#) \*parent, int32\_t value, int32\_t frame, int32\_t range, bool vertical=false)
- void [setVertical](#) (bool vertical, bool text=true)  
*vertical flag*
- bool **isHorizontal** () const
- bool **isVertical** () const
- void [setPrevText](#) (const char \*text)  
*scroll previous text*
- void **setPrevText** (const [String](#) &text)
- [String](#) **getPrevText** () const
- void [setNextText](#) (const char \*text)  
*scroll next text*
- void **setNextText** (const [String](#) &text)
- [String](#) **getNextText** () const
- void [setScrollColor](#) (const [Color](#) &color)  
*scroll color*
- const [Color](#) & **getScrollColor** () const
- void [setStep](#) (float64\_t step)  
*scroll step*
- float64\_t **getStep** () const

- void **setValue** (float64\_t value, bool callback=false)  
*scroll value*
- float64\_t **getValue** () const
- void **setFrame** (float64\_t frame)  
*scroll frame*
- float64\_t **getFrame** () const
- void **setRange** (float64\_t range)  
*scroll range*
- float64\_t **getRange** () const
- void **setFrameAlign** (Align align)  
*frame alignment*
- Align **getFrameAlign** () const
- bool **hasFrameAlign** (Align align) const
- bool **hasFrameAligns** (Align aligns) const
- void **setClickedCallback** (const ClickedCallback &func)
- ClickedCallback **getClickedCallback** () const
- bool **isClicked** ()
- void **setChangedCallback** (const ChangedCallback &func)
- ChangedCallback **getChangedCallback** () const
- bool **isChanged** (bool clear=true)
- CanvasMesh **getCanvasMesh** ()  
*canvas elements*

#### Additional Inherited Members

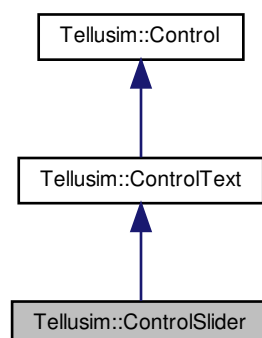
##### 5.55.1 Detailed Description

[ControlScroll](#) class

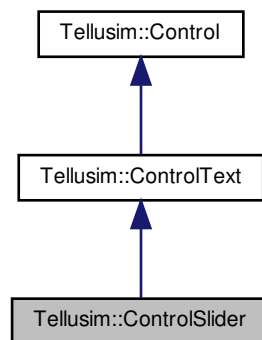
## 5.56 Tellusim::ControlSlider Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlSlider:



Collaboration diagram for Tellusim::ControlSlider:



### Public Types

- using [FormatCallback](#) = Function< [String](#)(ControlSlider)>  
*format callback*
- using [ReleasedCallback](#) = Function< void(ControlSlider)>  
*released callback*
- using [ClickedCallback](#) = Function< void(ControlSlider)>  
*clicked callback*
- using [ChangedCallback](#) = Function< void(ControlSlider)>  
*changed callback*

### Public Member Functions

- **ControlSlider** ([Control](#) \*parent)
- **ControlSlider** ([Control](#) \*parent, const char \*text)
- **ControlSlider** ([Control](#) \*parent, const [String](#) &text)
- **ControlSlider** ([Control](#) \*parent, const char \*text, uint32\_t digits)
- **ControlSlider** ([Control](#) \*parent, const char \*text, uint32\_t digits, float64\_t value)
- **ControlSlider** ([Control](#) \*parent, const char \*text, uint32\_t digits, float64\_t value, float64\_t min, float64\_t max)
- **ControlSlider** ([Control](#) \*parent, const char \*text, float64\_t value, float64\_t min, float64\_t max)
- **ControlSlider** ([Control](#) \*parent, const char \*text, uint32\_t value, uint32\_t min, uint32\_t max)
- **ControlSlider** ([Control](#) \*parent, const char \*text, int32\_t value, int32\_t min, int32\_t max)
- void [setSliderColor](#) (const [Color](#) &color)  
*slider color*
- const [Color](#) & [getSliderColor](#) () const
- void [setDigits](#) (uint32\_t digits)  
*slider digits*
- uint32\_t [getDigits](#) () const
- void [setTextEnabled](#) (bool enabled)  
*text enabled flag*
- bool [isTextEnabled](#) () const

- void [setStep](#) (float64\_t step)
  - slider step*
- float64\_t **getStep** () const
- void [setBase](#) (float64\_t base)
  - exponent base*
- float64\_t **getBase** () const
- void [setFormat](#) (const char \*format)
  - slider format*
- void **setFormat** (const [String](#) &format)
- [String](#) **getFormat** () const
- void [setValue](#) (float64\_t value, bool callback=false, bool exponent=false)
- float64\_t **getValue** (bool exponent=false) const
- float32\_t **getValuef32** (bool exponent=false) const
- uint32\_t **getValueu32** (bool exponent=false) const
- int32\_t **getValuei32** (bool exponent=false) const
- void [setRange](#) (float64\_t min, float64\_t max, bool exponent=false)
- float64\_t **getMinRange** (bool exponent=false) const
- float64\_t **getMaxRange** (bool exponent=false) const
- void [setHandleSize](#) (float32\_t size)
  - handle size*
- float32\_t **getHandleSize** () const
- void **setFormatCallback** (const [FormatCallback](#) &func)
- [FormatCallback](#) **getFormatCallback** () const
- void **setReleasedCallback** (const [ReleasedCallback](#) &func)
- [ReleasedCallback](#) **getReleasedCallback** () const
- bool **isReleased** ()
- void **setClickedCallback** (const [ClickedCallback](#) &func)
- [ClickedCallback](#) **getClickedCallback** () const
- bool **isClicked** ()
- void **setChangedCallback** (const [ChangedCallback](#) &func)
- [ChangedCallback](#) **getChangedCallback** () const
- bool **isChanged** (bool clear=true)
- [CanvasMesh](#) **getCanvasMesh** ()
  - canvas elements*

## Additional Inherited Members

### 5.56.1 Detailed Description

[ControlSlider](#) class

### 5.56.2 Member Function Documentation

#### 5.56.2.1 setValue()

```
void Tellusim::ControlSlider::setValue (
    float64_t value,
    bool callback = false,
    bool exponent = false )
```

slider value

## Parameters

|                 |                                       |
|-----------------|---------------------------------------|
| <i>callback</i> | Run changed callback on value change. |
| <i>exponent</i> | perform exponential conversion.       |

## 5.56.2.2 setRange()

```
void Tellusim::ControlSlider::setRange (
    float64_t min,
    float64_t max,
    bool exponent = false )
```

slider range

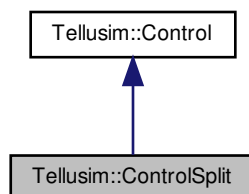
## Parameters

|                 |                                 |
|-----------------|---------------------------------|
| <i>exponent</i> | perform exponential conversion. |
|-----------------|---------------------------------|

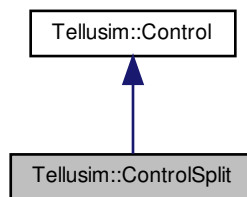
## 5.57 Tellusim::ControlSplit Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlSplit:



Collaboration diagram for Tellusim::ControlSplit:



#### Public Member Functions

- **ControlSplit** ([Control](#) \*parent, bool vertical=false)
- **ControlSplit** ([Control](#) \*parent, float32\_t value, bool vertical=false)
- void [setAbsolute](#) (bool absolute)  
*absolute flag*
- bool **isAbsolute** () const
- void [setVertical](#) (bool vertical)  
*vertical flag*
- bool **isHorizontal** () const
- bool **isVertical** () const
- void [setValue](#) (float32\_t value)  
*split value*
- float32\_t **getValue** () const
- void [setHandleSize](#) (float32\_t size)  
*handle size*
- float32\_t **getHandleSize** () const
- const [Vector2f](#) & [getControlsSize](#) () const  
*controls size*

#### Additional Inherited Members

#### 5.57.1 Detailed Description

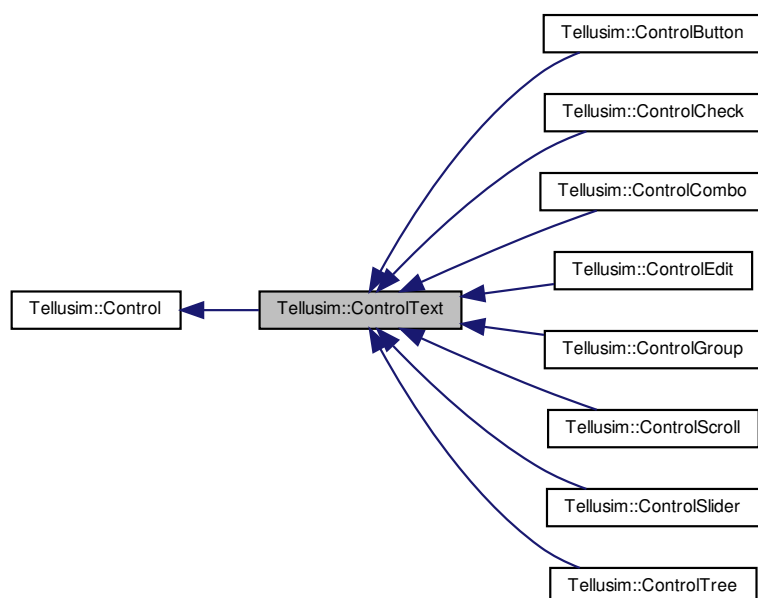
[ControlSplit](#) class



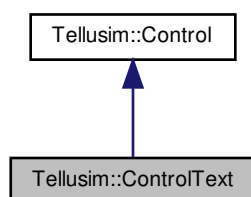
## 5.58 Tellusim::ControlText Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlText:



Collaboration diagram for Tellusim::ControlText:



### Public Member Functions

- **ControlText** ([Control](#) \*parent)
- **ControlText** ([Control](#) \*parent, const char \*text)
- **ControlText** ([Control](#) \*parent, const [String](#) &text)
- void **setMode** ([CanvasElement::Mode](#) mode)

- control mode*
  - [CanvasElement::Mode](#) **getMode** () const
  - void **setPipeline** ([Pipeline](#) &pipeline)
- control pipeline*
  - void **setPipeline** ([Pipeline](#) &pipeline, const [CanvasElement::DrawCallback](#) &func)
  - [Pipeline](#) **getPipeline** () const
  - void **setColor** (const [Color](#) &color)
- control color*
  - void **setColor** (float32\_t r, float32\_t g, float32\_t b, float32\_t a)
  - const [Color](#) & **getColor** () const
  - void **setFilter** ([Sampler::Filter](#) filter)
- filter mode*
  - [Sampler::Filter](#) **getFilter** () const
  - void **setAnisotropy** (uint32\_t anisotropy)
  - uint32\_t **getAnisotropy** () const
  - void **setBlend** ([Pipeline::BlendOp](#) op, [Pipeline::BlendFunc](#) src, [Pipeline::BlendFunc](#) dest)
- blending parameters*
  - [Pipeline::BlendOp](#) **getBlendOp** () const
  - [Pipeline::BlendFunc](#) **getBlendSrcFunc** () const
  - [Pipeline::BlendFunc](#) **getBlendDestFunc** () const
  - void **setFontName** (const char \*name)
- font name*
  - void **setFontName** (const [String](#) &name)
  - [String](#) **getFontName** () const
  - void **setFontColor** (const [Color](#) &color)
- font color*
  - const [Color](#) & **getFontColor** () const
  - bool **setFontSize** (uint32\_t size)
- font style*
  - uint32\_t **getFontSize** () const
  - bool **setFontStyle** (const [FontStyle](#) &style)
  - const [FontStyle](#) & **getFontStyleConst** () const
  - const [FontStyle](#) & **getFontStyle** () const
  - [FontStyle](#) & **getFontStyle** ()
  - void **setFontAlign** ([Align](#) align)
- font alignment*
  - [Align](#) **getFontAlign** () const
  - bool **hasFontAlign** ([Align](#) align) const
  - bool **hasFontAligns** ([Align](#) aligns) const
  - void **setText** (const char \*text)
- control text*
  - void **setText** (const [String](#) &text)
  - [String](#) **getText** () const
  - [CanvasText](#) **getCanvasText** ()
- canvas elements*

## Additional Inherited Members

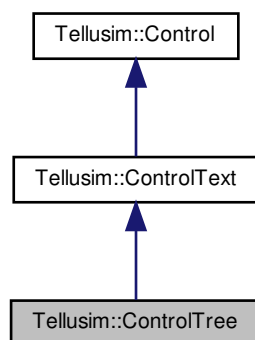
### 5.58.1 Detailed Description

[ControlText](#) class

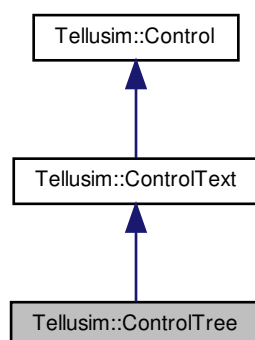
## 5.59 Tellusim::ControlTree Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlTree:



Collaboration diagram for Tellusim::ControlTree:



### Public Types

- using [ChangedCallback](#) = Function< void([ControlTree](#), uint32\_t item)>  
*changed callback*
- using [DraggedCallback](#) = Function< bool([ControlTree](#), uint32\_t item)>  
*dragged callback*
- using [DroppedCallback](#) = Function< void([ControlTree](#), uint32\_t item)>  
*dropped callback*

- using `ClickedCallback` = `Function< void(ControlTree, uint32_t item)>`  
*clicked callback*
- using `ExpandedCallback` = `Function< void(ControlTree, uint32_t item)>`  
*expanded callback*
- using `SelectedCallback` = `Function< void(ControlTree)>`  
*selected callback*

### Public Member Functions

- **ControlTree** (`Control` \*parent)
- void `setSelectable` (bool selectable)  
*selectable flag*
- bool `isSelectable` () const
- void `setMultiSelection` (bool multi\_selection)  
*multi-selection flag*
- bool `isMultiSelection` () const
- void `setFoldedText` (const char \*text)  
*folded text*
- void `setFoldedText` (const `String` &text)
- `String` `getFoldedText` () const
- void `setExpandedText` (const char \*text)  
*expanded text*
- void `setExpandedText` (const `String` &text)
- `String` `getExpandedText` () const
- void `setTexture` (`Texture` &texture, uint32\_t rows=1, uint32\_t columns=1)  
*texture pointer*
- `Texture` `getTexture` () const
- void `setTextureName` (const char \*name, uint32\_t rows=1, uint32\_t columns=1)  
*texture name*
- void `setTextureName` (const `String` &name, uint32\_t rows=1, uint32\_t columns=1)
- `String` `getTextureName` () const
- void `setTextureGrid` (uint32\_t rows, uint32\_t columns)  
*texture layout*
- uint32\_t `getTextureRows` () const
- uint32\_t `getTextureColumns` () const
- void `clearItems` ()  
*tree items*
- uint32\_t `addItem` (const char \*text, uint32\_t parent=Maxu32, bool expanded=true)
- uint32\_t `addItem` (const `String` &text, uint32\_t parent=Maxu32, bool expanded=true)
- void `addItems` (const `InitializerList`< const char \*> &items, uint32\_t parent=Maxu32)
- void `removeItem` (uint32\_t item, bool children=false)
- void `viewItem` (uint32\_t item)
- uint32\_t `getNumItems` () const
- uint32\_t `getItem` (uint32\_t index) const
- bool `switchItemHidden` (uint32\_t item, bool children=false)  
*item hidden flag*
- void `setItemHidden` (uint32\_t item, bool hidden, bool children=false)
- bool `isItemHidden` (uint32\_t item) const
- bool `switchItemExpanded` (uint32\_t item, bool children=false)  
*item expanded flag*
- void `setItemExpanded` (uint32\_t item, bool expanded, bool children=false)
- bool `isItemExpanded` (uint32\_t item) const

- bool **switchItemSelected** (uint32\_t item, bool children=false)  
*item selected flag*
- void **setItemSelected** (uint32\_t item, bool selected, bool children=false)
- bool **isItemSelected** (uint32\_t item) const
- void **setItemParent** (uint32\_t item, uint32\_t parent)  
*item parent*
- uint32\_t **getItemParent** (uint32\_t item) const
- bool **isItemParent** (uint32\_t item, uint32\_t parent, bool hierarchy=false) const
- void **addItemChild** (uint32\_t item, uint32\_t child)  
*item children*
- void **removeItemChild** (uint32\_t item, uint32\_t child)
- void **addItemChildren** (uint32\_t item, const Array< uint32\_t > &children)
- void **removeItemChildren** (uint32\_t item, const Array< uint32\_t > &children)
- uint32\_t **findItemChild** (uint32\_t item, uint32\_t child) const
- bool **isItemChild** (uint32\_t item, uint32\_t child) const
- uint32\_t **getNumItemChildren** (uint32\_t item) const
- uint32\_t **getItemChild** (uint32\_t item, uint32\_t index) const
- Array< uint32\_t > **getItemChildren** (uint32\_t item) const
- void **setItemText** (uint32\_t item, const char \*text)  
*item text*
- void **setItemText** (uint32\_t item, const **String** &text)
- **String** **getItemText** (uint32\_t item) const
- uint32\_t **findItemText** (const char \*text) const
- uint32\_t **findItemText** (const **String** &text) const
- void **setItemColor** (uint32\_t item, const **Color** &color)  
*item color*
- const **Color** & **getItemColor** (uint32\_t item) const
- void **setItemTexture** (uint32\_t item, uint32\_t row, uint32\_t column=0)  
*item icon*
- uint32\_t **getItemTextureRow** (uint32\_t item) const
- uint32\_t **getItemTextureColumn** (uint32\_t item) const
- void **setItemUserData** (uint32\_t item, void \*data)  
*item data*
- void \* **getItemUserData** (uint32\_t item) const
- void **setCurrentItem** (uint32\_t item, bool select=false, bool view=false, bool callback=false)  
*current item*
- uint32\_t **getCurrentItem** () const
- **String** **getCurrentText** () const
- void **setSelection** ()  
*selected items*
- void **clearSelection** ()
- void **inverseSelection** ()
- uint32\_t **getNumSelectedItems** () const
- uint32\_t **getSelectedItem** (uint32\_t index) const
- Array< uint32\_t > **getSelectedItems** () const
- void **setChangedCallback** (const **ChangedCallback** &func)
- **ChangedCallback** **getChangedCallback** () const
- void **setDraggedCallback** (const **DraggedCallback** &func)
- **DraggedCallback** **getDraggedCallback** () const
- void **setDroppedCallback** (const **DroppedCallback** &func)
- **DroppedCallback** **getDroppedCallback** () const
- void **setClickedCallback** (const **ClickedCallback** &func)
- void **setClicked2Callback** (const **ClickedCallback** &func)

- void **setClickedRightCallback** (const [ClickedCallback](#) &func)
- [ClickedCallback](#) **getClickedCallback** () const
- [ClickedCallback](#) **getClicked2Callback** () const
- [ClickedCallback](#) **getClickedRightCallback** () const
- void **setExpandedCallback** (const [ExpandedCallback](#) &func)
- [ExpandedCallback](#) **getExpandedCallback** () const
- void **setSelectedCallback** (const [SelectedCallback](#) &func)
- [SelectedCallback](#) **getSelectedCallback** () const

#### Additional Inherited Members

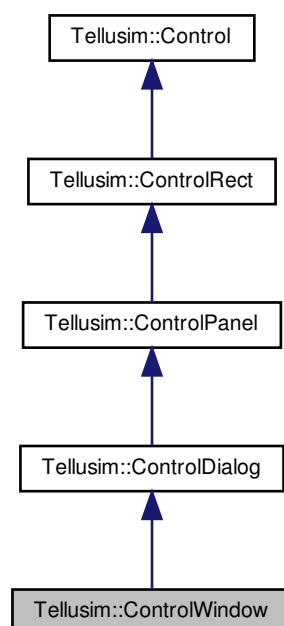
##### 5.59.1 Detailed Description

[ControlTree](#) class

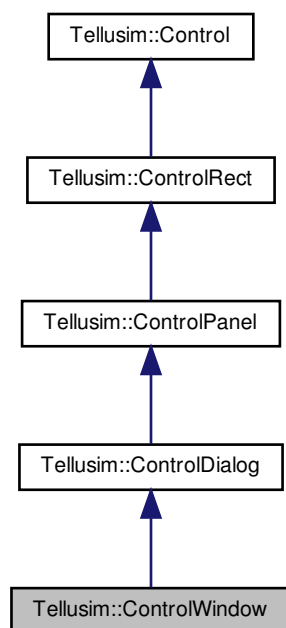
## 5.60 Tellusim::ControlWindow Class Reference

```
#include <interface/TellusimControls.h>
```

Inheritance diagram for Tellusim::ControlWindow:



Collaboration diagram for Tellusim::ControlWindow:



#### Public Member Functions

- **ControlWindow** ([ControlRoot](#) \*root, [Window](#) &parent, [Window](#) &window)
- **ControlWindow** ([ControlRoot](#) \*root, [Window](#) &parent, [Window](#) &window, uint32\_t columns)
- **ControlWindow** ([ControlRoot](#) \*root, [Window](#) &parent, [Window](#) &window, uint32\_t columns, float32\_t x, float32\_t y)
- [Window](#) getParentWindow () const  
*control windows*
- [Window](#) getDialogWindow () const

#### Additional Inherited Members

##### 5.60.1 Detailed Description

[ControlWindow](#) class

## 5.61 Tellusim::CubeFilter Class Reference

```
#include <graphics/TellusimCubeFilter.h>
```

## Public Types

- enum [Mode](#) {  
**ModeCube** = 0,  
**ModePanorama**,  
**NumModes** }  
*Filter modes.*
- enum [Flags](#) {  
**FlagCube** = (1 << ModeCube),  
**FlagPanorama** = (1 << ModePanorama),  
**FlagsAll** = (FlagCube | FlagPanorama) }  
*Filter flags.*

## Public Member Functions

- void [clear](#) ()  
*clear filter*
- bool [isCreated](#) ([Mode](#) mode) const  
*check filter*
- uint32\_t [getGroupSize](#) () const  
*filter parameters*
- uint32\_t [getMaxOrder](#) () const
- uint32\_t [getMaxSize](#) () const
- uint32\_t [getHarmonics](#) () const
- bool [create](#) (const [Device](#) &device, [Mode](#) mode, uint32\_t order=3, uint32\_t size=1024, uint32\_t groups=256)
- bool [create](#) (const [Device](#) &device, [Flags](#) flags, uint32\_t order=3, uint32\_t size=1024, uint32\_t groups=256)
- bool [dispatch](#) ([Compute](#) &compute, [Buffer](#) &buffer, uint32\_t offset, [Texture](#) &texture, const [Slice](#) &slice) const
- bool [dispatch](#) ([Compute](#) &compute, [Buffer](#) &buffer, uint32\_t offset, [Texture](#) &texture) const
- bool [dispatch](#) ([Compute](#) &compute, [Texture](#) &texture, const [Slice](#) &slice, [Buffer](#) &buffer, uint32\_t offset) const
- bool [dispatch](#) ([Compute](#) &compute, [Texture](#) &texture, [Buffer](#) &buffer, uint32\_t offset) const

## 5.61.1 Detailed Description

[CubeFilter](#) class

## 5.61.2 Member Function Documentation

5.61.2.1 [create\(\)](#)

```
bool Tellusim::CubeFilter::create (
    const Device & device,
    Mode mode,
    uint32_t order = 3,
    uint32_t size = 1024,
    uint32_t groups = 256 )
```

create filter



## Parameters

|               |                                 |
|---------------|---------------------------------|
| <i>order</i>  | Maximum filter order (2, 3, 5). |
| <i>size</i>   | Maximum filter size.            |
| <i>groups</i> | Reduction group size.           |

## 5.61.2.2 dispatch() [1/2]

```
bool Tellusim::CubeFilter::dispatch (
    Compute & compute,
    Buffer & buffer,
    uint32_t offset,
    Texture & texture,
    const Slice & slice ) const
```

dispatch cube filter

## Parameters

|                |                                  |
|----------------|----------------------------------|
| <i>buffer</i>  | Destination coefficients buffer. |
| <i>offset</i>  | Destination buffer offset.       |
| <i>texture</i> | Source cube texture.             |
| <i>slice</i>   | Source texture slice.            |

## 5.61.2.3 dispatch() [2/2]

```
bool Tellusim::CubeFilter::dispatch (
    Compute & compute,
    Texture & texture,
    const Slice & slice,
    Buffer & buffer,
    uint32_t offset ) const
```

dispatch cube render

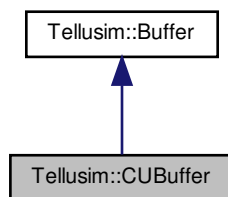
## Parameters

|                |                                     |
|----------------|-------------------------------------|
| <i>texture</i> | Destination cube texture (RGBAf16). |
| <i>slice</i>   | Destination texture slice.          |
| <i>buffer</i>  | Source coefficients buffer.         |
| <i>offset</i>  | Source buffer offset.               |

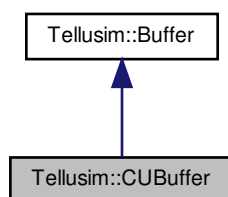
## 5.62 Tellusim::CUBuffer Class Reference

```
#include <platform/TellusimBuffer.h>
```

Inheritance diagram for Tellusim::CUBuffer:



Collaboration diagram for Tellusim::CUBuffer:



#### Public Member Functions

- `size_t getBufferPtr () const`
- `uint8_t * getBufferData () const`
- `CUevent getBufferEvent () const`
- `uint32_t getArrayFormat () const`
- `uint32_t getArrayChannels () const`
- `CUexternalMemory getSharedMemory () const`

#### Additional Inherited Members

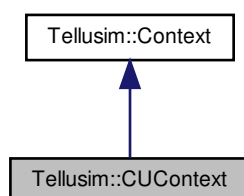
##### 5.62.1 Detailed Description

#### CUBuffer

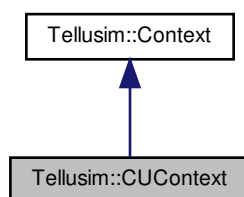
## 5.63 Tellusim::CUContext Class Reference

```
#include <platform/TellusimContext.h>
```

Inheritance diagram for Tellusim::CUContext:



Collaboration diagram for Tellusim::CUContext:



### Public Member Functions

- `int32_t` [getDevice](#) () const  
*current device*
- `CUcontext` **getCUContext** () const
- `CUstream` **getStream** () const

### Static Public Member Functions

- `static void *` [getProcAddress](#) (const char \*name)  
*Cuda functions.*
- `static bool` [error](#) (uint32\_t result)  
*check Cuda errors*

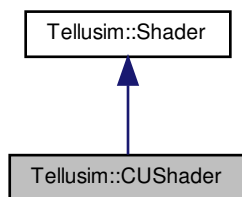
### 5.63.1 Detailed Description

[CUContext](#)

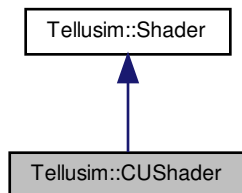
## 5.64 Tellusim::CShader Class Reference

```
#include <platform/TellusimShader.h>
```

Inheritance diagram for Tellusim::CShader:



Collaboration diagram for Tellusim::CShader:



### Public Member Functions

- CUmodule **getModule** () const
- CUfunction **getFunction** () const

### Additional Inherited Members

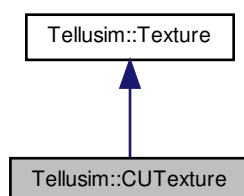
### 5.64.1 Detailed Description

[CShader](#)

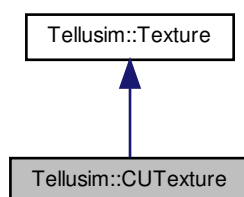
## 5.65 Tellusim::CUTexture Class Reference

```
#include <platform/TellusimTexture.h>
```

Inheritance diagram for Tellusim::CUTexture:



Collaboration diagram for Tellusim::CUTexture:



### Public Member Functions

- CUmipmappedArray **getTextureArray** () const
- CUarray **getTextureLevel** (uint32\_t index) const
- uint32\_t **getArrayFormat** () const
- uint32\_t **getArrayChannels** () const
- CUexternalMemory **getSharedMemory** () const

### Additional Inherited Members

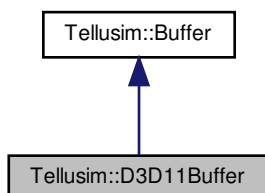
#### 5.65.1 Detailed Description

#### CUTexture

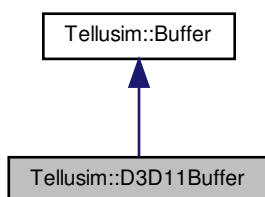
## 5.66 Tellusim::D3D11Buffer Class Reference

```
#include <platform/TellusimBuffer.h>
```

Inheritance diagram for Tellusim::D3D11Buffer:



Collaboration diagram for Tellusim::D3D11Buffer:



### Public Member Functions

- bool **create** ([Flags](#) flags, ID3D11Buffer \*buffer)  
*create external buffer*
- ID3D11Buffer \* **getD3D11Buffer** () const
- ID3D11UnorderedAccessView \* **getUnorderedAccessView** () const
- ID3D11ShaderResourceView \* **getShaderResourceView** () const
- void \* **getInteropHandle** () const

### Additional Inherited Members

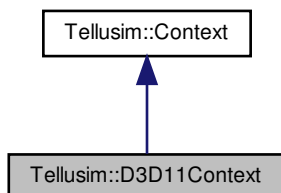
#### 5.66.1 Detailed Description

[D3D11Buffer](#)

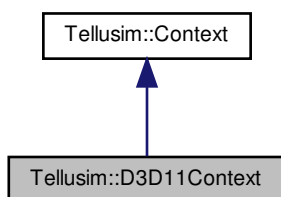
## 5.67 Tellusim::D3D11Context Class Reference

```
#include <platform/TellusimContext.h>
```

Inheritance diagram for Tellusim::D3D11Context:



Collaboration diagram for Tellusim::D3D11Context:



### Public Member Functions

- bool [create](#) (ID3D11Device \*device)  
*create context*
- IDXGIFactory \* [getFactory](#) () const  
*current device*
- ID3D11Device \* [getDevice](#) () const
- ID3D11DeviceContext \* [getD3D11Context](#) () const

### Static Public Member Functions

- static void \* [getProcAddress](#) (const char \*name)  
*Direct3D11 functions.*
- static bool [error](#) (uint32\_t result)  
*check Direct3D11 errors*

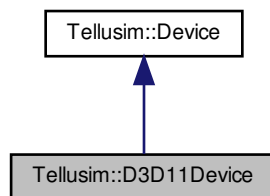
### 5.67.1 Detailed Description

[D3D11Context](#)

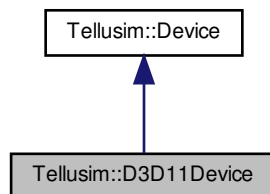
## 5.68 Tellusim::D3D11Device Class Reference

```
#include <platform/TellusimDevice.h>
```

Inheritance diagram for Tellusim::D3D11Device:



Collaboration diagram for Tellusim::D3D11Device:



### Public Member Functions

- **D3D11Device** ([Context](#) &context)
- **D3D11Device** ([Surface](#) &surface)
- **D3D11Device** ([Window](#) &window)
- ID3D11Device \* [getD3D11Device](#) () const  
*command context*
- ID3D11DeviceContext \* **getCommand** () const

### 5.68.1 Detailed Description

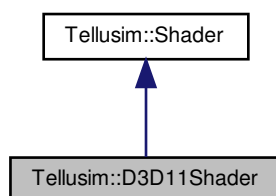
[D3D11Device](#)



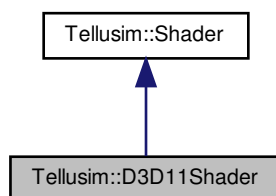
## 5.69 Tellusim::D3D11Shader Class Reference

```
#include <platform/TellusimShader.h>
```

Inheritance diagram for Tellusim::D3D11Shader:



Collaboration diagram for Tellusim::D3D11Shader:



### Public Member Functions

- void \* **getD3D11Shader** () const
- ID3DBlob \* **getShaderBlob** () const

### Additional Inherited Members

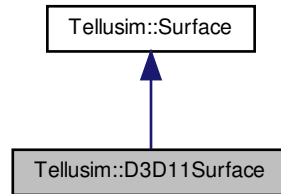
#### 5.69.1 Detailed Description

[D3D11Shader](#)

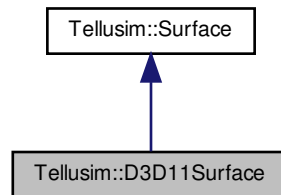
## 5.70 Tellusim::D3D11Surface Class Reference

```
#include <platform/TellusimSurface.h>
```

Inheritance diagram for Tellusim::D3D11Surface:



Collaboration diagram for Tellusim::D3D11Surface:



### Public Member Functions

- **D3D11Surface** ([D3D11Context](#) &context)
- [IDXGIFactory](#) \* [getFactory](#) () const  
*current device*
- [ID3D11Device](#) \* [getDevice](#) () const
- [ID3D11DeviceContext](#) \* [getContext](#) () const
- void [setSwapChain](#) ([IDXGISwapChain](#) \*swap\_chain)  
*swap chain*
- [IDXGISwapChain](#) \* [getSwapChain](#) () const
- void [setRenderTarget](#) ([ID3D11Texture2D](#) \*render\_target)  
*render targets*
- void [setDepthStencil](#) ([ID3D11Texture2D](#) \*depth\_stencil)
- [ID3D11Texture2D](#) \* [getRenderTarget](#) () const
- [ID3D11Texture2D](#) \* [getDepthStencil](#) () const
- void [setRenderTargetView](#) ([ID3D11RenderTargetView](#) \*render\_target\_view)  
*render target views*

- void **setDepthStencilView** (ID3D11DepthStencilView \*depth\_stencil\_view)
- ID3D11RenderTargetView \* **getRenderTargetView** () const
- ID3D11DepthStencilView \* **getDepthStencilView** () const
- uint32\_t **getColorDXGIFormat** () const  
*surface formats*
- uint32\_t **getDepthDXGIFormat** () const

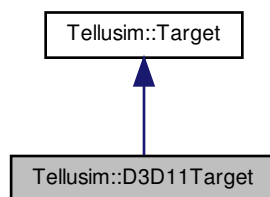
#### 5.70.1 Detailed Description

#### D3D11Surface

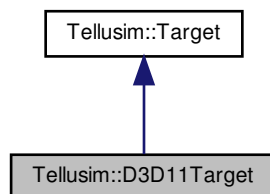
### 5.71 Tellusim::D3D11Target Class Reference

```
#include <platform/TellusimTarget.h>
```

Inheritance diagram for Tellusim::D3D11Target:



Collaboration diagram for Tellusim::D3D11Target:



#### Public Member Functions

- ID3D11RenderTargetView \*\* **getRenderTargetViews** () const
- ID3D11DepthStencilView \* **getDepthStencilView** () const

## Additional Inherited Members

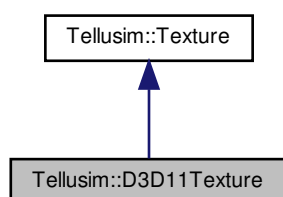
### 5.71.1 Detailed Description

#### [D3D11Target](#)

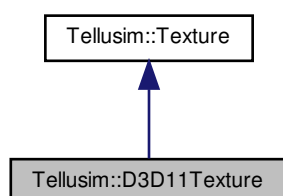
## 5.72 Tellusim::D3D11Texture Class Reference

```
#include <platform/TellusimTexture.h>
```

Inheritance diagram for Tellusim::D3D11Texture:



Collaboration diagram for Tellusim::D3D11Texture:



## Public Member Functions

- bool [create](#) (Type type, ID3D11Texture2D \*texture, [Flags](#) flags=DefaultFlags, Format format=FormatUnknown)  
*create external texture*
- uint32\_t **getDXGIFormat** () const
- ID3D11Texture2D \* **getD3D11Texture** () const
- ID3D11ShaderResourceView \* **getShaderResourceView** () const
- ID3D11RenderTargetView \* **getRenderTargetView** () const
- ID3D11DepthStencilView \* **getDepthStencilView** () const
- ID3D11UnorderedAccessView \* **getUnorderedAccessView** () const
- void \* **getInteropHandle** () const

## Additional Inherited Members

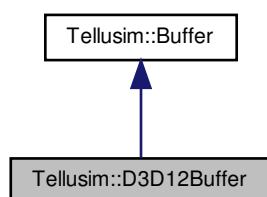
## 5.72.1 Detailed Description

[D3D11Texture](#)

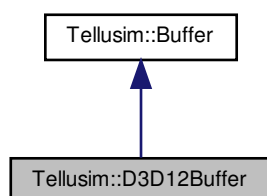
## 5.73 Tellusim::D3D12Buffer Class Reference

```
#include <platform/TellusimBuffer.h>
```

Inheritance diagram for Tellusim::D3D12Buffer:



Collaboration diagram for Tellusim::D3D12Buffer:



## Public Member Functions

- `bool` **create** (`Flags` flags, `ID3D12Resource *`buffer, `uint32_t` state)  
*create external buffer*
- `ID3D12Resource *` **getD3D12Buffer** () const
- `size_t` **getUnorderedAccessView** () const
- `size_t` **getShaderResourceView** () const
- `uint64_t` **getBufferAddress** () const
- `void` **setBufferState** (`uint32_t` state)
- `uint32_t` **getBufferState** () const
- `void *` **getSharedHandle** () const
- `void *` **getInteropHandle** () const

## Additional Inherited Members

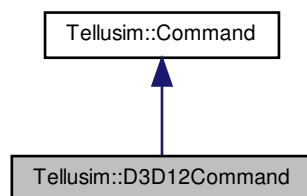
### 5.73.1 Detailed Description

#### [D3D12Buffer](#)

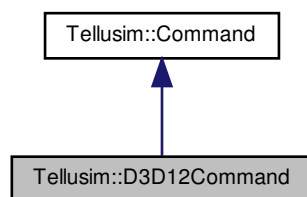
## 5.74 Tellusim::D3D12Command Class Reference

```
#include <platform/TellusimCommand.h>
```

Inheritance diagram for Tellusim::D3D12Command:



Collaboration diagram for Tellusim::D3D12Command:



## Public Member Functions

- ID3D12GraphicsCommandList \* [getD3D12Command](#) () const  
*command context*
- void [update](#) ()  
*update resources*

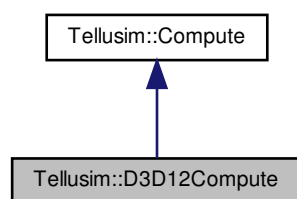
## 5.74.1 Detailed Description

[D3D12Command](#)

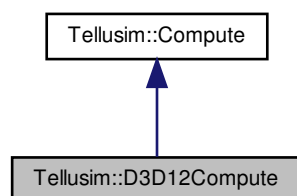
## 5.75 Tellusim::D3D12Compute Class Reference

```
#include <platform/TellusimCompute.h>
```

Inheritance diagram for Tellusim::D3D12Compute:



Collaboration diagram for Tellusim::D3D12Compute:

**Public Member Functions**

- ID3D12GraphicsCommandList \* [getCommand](#) () const  
*command context*
- void [update](#) ()  
*update resources*

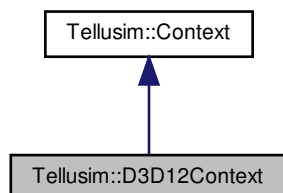
## 5.75.1 Detailed Description

[D3D12Compute](#)

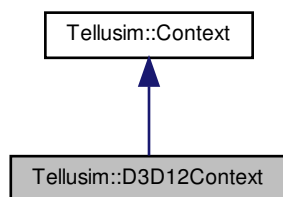
## 5.76 Tellusim::D3D12Context Class Reference

```
#include <platform/TellusimContext.h>
```

Inheritance diagram for Tellusim::D3D12Context:



Collaboration diagram for Tellusim::D3D12Context:



### Public Member Functions

- bool [create](#) (ID3D12Device \*device, ID3D12CommandQueue \*queue)  
*create context*
- IDXGIFactory4 \* [getFactory](#) () const  
*current device*
- ID3D12Device \* **getDevice** () const
- ID3D12CommandQueue \* **getQueue** () const
- ID3D12GraphicsCommandList \* **getCommand** () const

### Static Public Member Functions

- static void \* [getProcAddress](#) (const char \*name)  
*Direct3D12 functions.*
- static bool [error](#) (uint32\_t result)  
*check Direct3D12 errors*



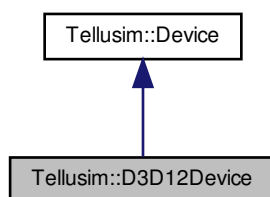
## 5.76.1 Detailed Description

[D3D12Context](#)

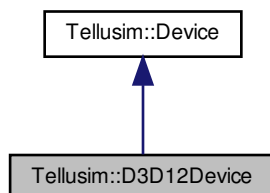
## 5.77 Tellusim::D3D12Device Class Reference

```
#include <platform/TellusimDevice.h>
```

Inheritance diagram for Tellusim::D3D12Device:



Collaboration diagram for Tellusim::D3D12Device:



## Public Member Functions

- **D3D12Device** ([Context](#) &context)
- **D3D12Device** ([Surface](#) &surface)
- **D3D12Device** ([Window](#) &window)
- void [setBufferState](#) ([Buffer](#) &buffer, uint32\_t state)  
    *buffer state*
- void [setTextureState](#) ([Texture](#) &texture, uint32\_t state)  
    *texture state*
- ID3D12Device \* [getD3D12Device](#) () const  
    *command context*
- ID3D12CommandQueue \* **getQueue** () const
- ID3D12GraphicsCommandList \* **getCommand** () const

### 5.77.1 Detailed Description

[D3D12Device](#)

## 5.78 Tellusim::D3D12Tracing::D3D12Instance Struct Reference

tracing instance

```
#include <platform/TellusimTracing.h>
```

### Public Attributes

- float32\_t **transform** [12]
- uint32\_t **data**: 24
- uint32\_t **mask**: 8
- uint32\_t **offset**: 24
- uint32\_t **flags**: 8
- uint64\_t **address**

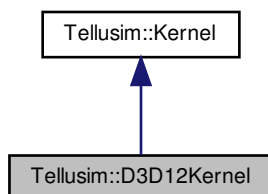
### 5.78.1 Detailed Description

tracing instance

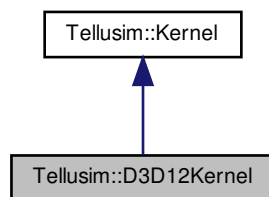
## 5.79 Tellusim::D3D12Kernel Class Reference

```
#include <platform/TellusimKernel.h>
```

Inheritance diagram for Tellusim::D3D12Kernel:



Collaboration diagram for Tellusim::D3D12Kernel:



#### Public Member Functions

- ID3D12RootSignature \* **getRootSignature** () const

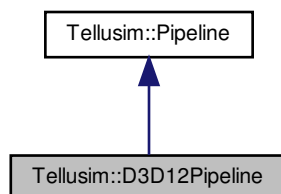
#### 5.79.1 Detailed Description

[D3D12Kernel](#)

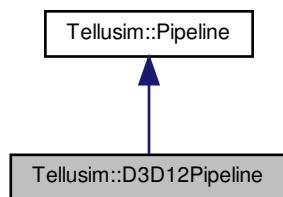
## 5.80 Tellusim::D3D12Pipeline Class Reference

```
#include <platform/TellusimPipeline.h>
```

Inheritance diagram for Tellusim::D3D12Pipeline:



Collaboration diagram for Tellusim::D3D12Pipeline:



#### Public Member Functions

- ID3D12RootSignature \* **getRootSignature** () const

#### Additional Inherited Members

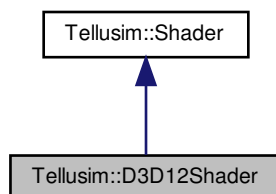
##### 5.80.1 Detailed Description

[D3D12Pipeline](#)

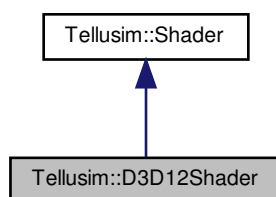
## 5.81 Tellusim::D3D12Shader Class Reference

```
#include <platform/TellusimShader.h>
```

Inheritance diagram for Tellusim::D3D12Shader:



Collaboration diagram for Tellusim::D3D12Shader:



#### Public Member Functions

- ID3DBlob \* **getShaderBlob** () const

#### Additional Inherited Members

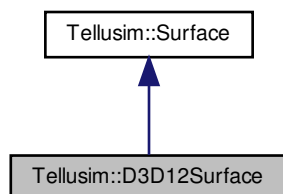
##### 5.81.1 Detailed Description

#### D3D12Shader

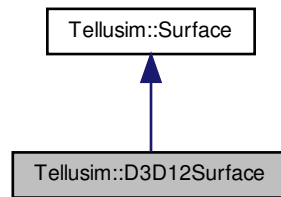
## 5.82 Tellusim::D3D12Surface Class Reference

```
#include <platform/TellusimSurface.h>
```

Inheritance diagram for Tellusim::D3D12Surface:



Collaboration diagram for Tellusim::D3D12Surface:



### Public Member Functions

- **D3D12Surface** ([D3D12Context](#) &context)
- IDXGIFactory4 \* [getFactory](#) () const  
*current device*
- ID3D12Device \* **getDevice** () const
- ID3D12CommandQueue \* **getQueue** () const
- ID3D12GraphicsCommandList \* **getCommand** () const
- void [setSwapChain](#) (IDXGISwapChain \*swap\_chain)  
*swap chain*
- IDXGISwapChain \* **getSwapChain** () const
- void [setRenderTarget](#) (ID3D12Resource \*render\_target)  
*render targets*
- void **setDepthStencil** (ID3D12Resource \*depth\_stencil)
- ID3D12Resource \* **getRenderTarget** () const
- ID3D12Resource \* **getDepthStencil** () const
- void [setRenderTargetView](#) (size\_t render\_target\_view)  
*render target views*
- void **setDepthStencilView** (size\_t depth\_stencil\_view)
- size\_t **getRenderTargetView** () const
- size\_t **getDepthStencilView** () const
- uint32\_t [getColorDXGIFormat](#) () const  
*surface formats*
- uint32\_t **getDepthDXGIFormat** () const

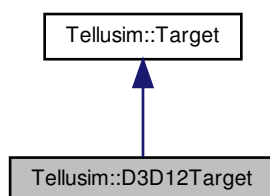
#### 5.82.1 Detailed Description

[D3D12Surface](#)

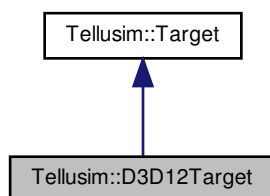
## 5.83 Tellusim::D3D12Target Class Reference

```
#include <platform/TellusimTarget.h>
```

Inheritance diagram for Tellusim::D3D12Target:



Collaboration diagram for Tellusim::D3D12Target:



### Public Member Functions

- `size_t * getRenderTargetViews () const`
- `size_t getDepthStencilView () const`

### Additional Inherited Members

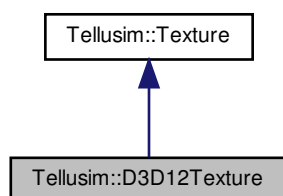
#### 5.83.1 Detailed Description

[D3D12Target](#)

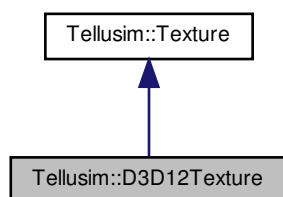
## 5.84 Tellusim::D3D12Texture Class Reference

```
#include <platform/TellusimTexture.h>
```

Inheritance diagram for Tellusim::D3D12Texture:



Collaboration diagram for Tellusim::D3D12Texture:



### Public Member Functions

- bool [create](#) (Type type, ID3D12Resource \*texture, uint32\_t state, [Flags](#) flags=DefaultFlags, Format format=FormatUnknown)  
*create external texture*
- uint32\_t [getDXGIFormat](#) () const
- ID3D12Resource \* [getD3D12Texture](#) () const
- size\_t [getShaderResourceView](#) () const
- size\_t [getRenderTargetView](#) () const
- size\_t [getDepthStencilView](#) () const
- size\_t [getUnorderedAccessView](#) () const
- void [setTextureState](#) (uint32\_t state)
- uint32\_t [getTextureState](#) () const
- void \* [getSharedHandle](#) () const
- void \* [getInteropHandle](#) () const



## Additional Inherited Members

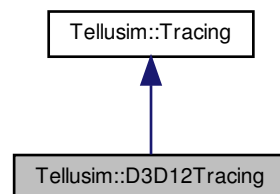
## 5.84.1 Detailed Description

[D3D12Texture](#)

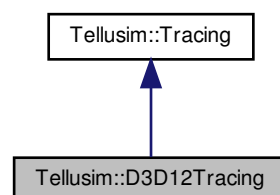
## 5.85 Tellusim::D3D12Tracing Class Reference

```
#include <platform/TellusimTracing.h>
```

Inheritance diagram for Tellusim::D3D12Tracing:



Collaboration diagram for Tellusim::D3D12Tracing:



## Classes

- struct [D3D12Instance](#)  
*tracing instance*

## Public Member Functions

- void \* **getGeometryDesc** (uint32\_t index) const
- void \* **getBuildInputs** () const
- void \* **getPrebuildInfo** () const
- void \* **getBuildDesc** () const
- [Buffer](#) **getTracingBuffer** () const
- size\_t **getShaderResourceView** () const

## Additional Inherited Members

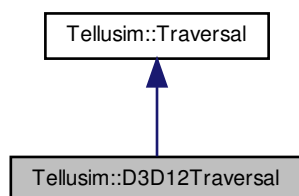
## 5.85.1 Detailed Description

[D3D12Tracing](#)

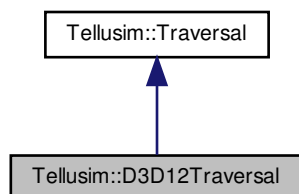
## 5.86 Tellusim::D3D12Traversal Class Reference

```
#include <platform/TellusimTraversal.h>
```

Inheritance diagram for Tellusim::D3D12Traversal:



Collaboration diagram for Tellusim::D3D12Traversal:



## Public Member Functions

- ID3D12RootSignature \* **getRootSignature** () const

## 5.86.1 Detailed Description

[D3D12Traversal](#)

## 5.87 Tellusim::Date Class Reference

```
#include <core/TellusimTime.h>
```

## Public Member Functions

- **Date** (int64\_t time, bool local=true)
- **Date** (const char \*str, const char \*format=nullptr)
- void [clear](#) ()  
*clear date*
- void [setTime](#) (int64\_t time, bool local=true)  
*time in seconds since 1970-01-01 00:00:00*
- int64\_t **getTime** (bool local=true) const
- bool [setString](#) (const char \*str, const char \*format=nullptr)
- [String](#) **getString** (const char \*format=nullptr) const
- void [setYear](#) (uint32\_t year)  
*current date*
- void **setMonth** (uint32\_t month)
- void **setDate** (uint32\_t date)
- void **setDay** (uint32\_t day)
- void **setHours** (uint32\_t hours)
- void **setMinutes** (uint32\_t minutes)
- void **setSeconds** (uint32\_t seconds)
- uint32\_t **getYear** () const
- uint32\_t **getMonth** () const
- uint32\_t **getDate** () const
- uint32\_t **getDay** () const
- uint32\_t **getHours** () const
- uint32\_t **getMinutes** () const
- uint32\_t **getSeconds** () const

## Static Public Member Functions

- static int32\_t [getTimeZone](#) ()  
*local timezone in seconds*

## 5.87.1 Detailed Description

[Date](#) class

## 5.87.2 Member Function Documentation

### 5.87.2.1 setString()

```
bool Tellusim::Date::setString (
    const char * str,
    const char * format = nullptr )
```

string time value

#### Parameters

|               |                                            |
|---------------|--------------------------------------------|
| <i>format</i> | Default time format is yyyy-MM-dd HH:mm:ss |
|---------------|--------------------------------------------|

## 5.88 Tellusim::DecoderJPEG Class Reference

```
#include <graphics/TellusimDecoderJPEG.h>
```

### Public Types

- enum [Mode](#) {  
**ModeR** = 0,  
**ModeRG**,  
**ModeRGBA**,  
**ModeYUV444**,  
**ModeYUV422H**,  
**ModeYUV422V**,  
**ModeYUV420**,  
**NumModes** }  
*Decoder modes.*
- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagR** = (1 << ModeR),  
**FlagRG** = (1 << ModeRG),  
**FlagRGBA** = (1 << ModeRGBA),  
**FlagYUV444** = (1 << ModeYUV444),  
**FlagYUV422H** = (1 << ModeYUV422H),  
**FlagYUV422V** = (1 << ModeYUV422V),  
**FlagYUV420** = (1 << ModeYUV420),  
**FlagsAll** = (FlagR | FlagRG | FlagRGBA | FlagYUV444 | FlagYUV422H | FlagYUV422V | FlagYUV420) }  
*Decoder flags.*

### Public Member Functions

- void [clear](#) ()  
*clear decoder*
- bool [isCreated](#) ([Mode](#) mode) const  
*check decoder*

- bool **create** (const [Device](#) &device, [Mode](#) mode)  
*create decoder*
- bool **create** (const [Device](#) &device, [Flags](#) flags)
- [Texture](#) **loadTexture** (const [Device](#) &device, const char \*name, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- [Texture](#) **loadTexture** (const [Device](#) &device, [Stream](#) &stream, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- bool **dispatch** ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src, const [Slice](#) &dest\_slice, const [Slice](#) &src\_slice) const
- bool **dispatch** ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src, const [Slice](#) &src\_slice) const
- bool **dispatch** ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src) const
- bool **dispatchYUV** ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src, const [Slice](#) &dest\_slice, const [Slice](#) &src\_slice) const
- bool **dispatchYUV** ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src, const [Slice](#) &src\_slice) const
- bool **dispatchYUV** ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src) const

### Static Public Member Functions

- static bool **isYUV** ([Mode](#) mode)  
*YUV444 mode performs inplace YUVtoRGB conversion.*
- static bool **load** (const char \*name, [Image](#) &image, [Mode](#) &mode, [Size](#) &size)
- static bool **load** ([Stream](#) &stream, [Image](#) &image, [Mode](#) &mode, [Size](#) &size)

#### 5.88.1 Detailed Description

[DecoderJPEG](#) class

#### 5.88.2 Member Function Documentation

##### 5.88.2.1 load()

```
static bool Tellusim::DecoderJPEG::load (
    const char * name,
    Image & image,
    Mode & mode,
    Size & size ) [static]
```

load decoder image

#### Parameters

|              |                            |
|--------------|----------------------------|
| <i>name</i>  | <a href="#">Image</a> name |
| <i>image</i> | Decoder image.             |
| <i>mode</i>  | Decoding mode.             |
| <i>size</i>  | Decoding size.             |

### 5.88.2.2 loadTexture()

```
Texture Tellusim::DecoderJPEG::loadTexture (
    const Device & device,
    const char * name,
    Texture::Flags flags = Texture::DefaultFlags ) const
```

load texture from image

#### Parameters

|               |                 |
|---------------|-----------------|
| <i>device</i> | Device pointer. |
| <i>name</i>   | Image name.     |
| <i>flags</i>  | Texture flags.  |

### 5.88.2.3 dispatch()

```
bool Tellusim::DecoderJPEG::dispatch (
    Compute & compute,
    Mode mode,
    Texture & dest,
    Texture & src,
    const Slice & dest_slice,
    const Slice & src_slice ) const
```

dispatch decoder

#### Parameters

|                   |                            |
|-------------------|----------------------------|
| <i>mode</i>       | Decoder mode.              |
| <i>dest</i>       | Destination surface.       |
| <i>src</i>        | Source FFT surface.        |
| <i>dest_slice</i> | Destination texture slice. |
| <i>src_slice</i>  | Source texture slice.      |

### 5.88.2.4 dispatchYUV()

```
bool Tellusim::DecoderJPEG::dispatchYUV (
    Compute & compute,
    Mode mode,
    Texture & dest,
    Texture & src,
    const Slice & dest_slice,
    const Slice & src_slice ) const
```

dispatch YUV converter

## Parameters

|                   |                                       |
|-------------------|---------------------------------------|
| <i>mode</i>       | Decoder mode.                         |
| <i>dest</i>       | Destination surface.                  |
| <i>src</i>        | <a href="#">Source</a> YUV surface.   |
| <i>dest_slice</i> | Destination texture slice.            |
| <i>src_slice</i>  | <a href="#">Source</a> texture slice. |

## 5.89 Tellusim::DefaultDestructor&lt; Type &gt; Struct Template Reference

```
#include <core/TellusimPointer.h>
```

## Static Public Member Functions

- static void [destructor](#) (Type \*ptr)  
*delete pointer*

## 5.89.1 Detailed Description

```
template<class Type>
struct Tellusim::DefaultDestructor< Type >
```

Default destructor

## 5.90 Tellusim::Desktop Class Reference

```
#include <system/TellusimDesktop.h>
```

## Public Member Functions

- bool [update](#) ()  
*update configuration*
- uint32\_t [getWidth](#) () const  
*desktop resolution*
- uint32\_t [getHeight](#) () const
- int32\_t [getPositionX](#) () const
- int32\_t [getPositionY](#) () const
- float32\_t [getScale](#) () const
- uint32\_t [getNumScreens](#) () const  
*screen configuration*
- [String](#) [getScreenName](#) (uint32\_t index) const
- [String](#) [getScreenDevice](#) (uint32\_t index) const
- uint32\_t [getScreenWidth](#) (uint32\_t index) const
- uint32\_t [getScreenHeight](#) (uint32\_t index) const
- int32\_t [getScreenPositionX](#) (uint32\_t index) const
- int32\_t [getScreenPositionY](#) (uint32\_t index) const

- uint32\_t **getScreenFrequency** (uint32\_t index) const
- uint32\_t **getNumModes** (uint32\_t index) const  
*screen video modes*
- uint32\_t **getModeWidth** (uint32\_t index, uint32\_t mode) const
- uint32\_t **getModeHeight** (uint32\_t index, uint32\_t mode) const
- uint32\_t **getModelIndex** (uint32\_t index, uint32\_t width, uint32\_t height) const  
*change screen resolution*
- bool **setMode** (uint32\_t index, uint32\_t width, uint32\_t height)
- bool **restoreMode** (uint32\_t index)
- uint32\_t **getWidth** (uint32\_t index) const  
*current screen configuration*
- uint32\_t **getHeight** (uint32\_t index) const
- int32\_t **getPositionX** (uint32\_t index) const
- int32\_t **getPositionY** (uint32\_t index) const
- uint32\_t **getScreenIndex** (int32\_t x, int32\_t y) const  
*get current screen index*
- bool **setMouse** (int32\_t x, int32\_t y) const  
*mouse position*
- bool **getMouse** (int32\_t &x, int32\_t &y) const

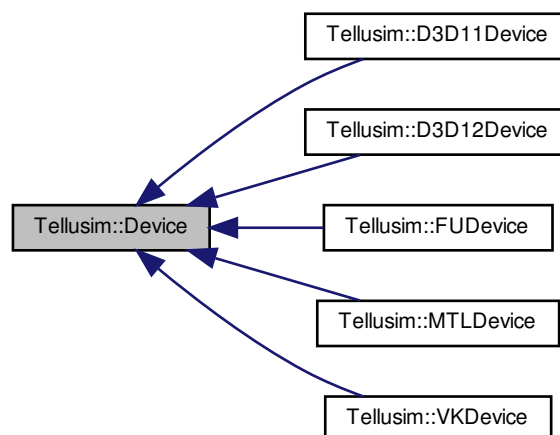
#### 5.90.1 Detailed Description

[Desktop](#) configuration

### 5.91 Tellusim::Device Class Reference

```
#include <platform/TellusimDevice.h>
```

Inheritance diagram for Tellusim::Device:





## Classes

- struct [Features](#)  
*device features*

## Public Member Functions

- **Device** ([Context](#) &context)
- **Device** ([Surface](#) &surface)
- **Device** ([Window](#) &window)
- Platform [getPlatform](#) () const  
*device platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*device index*
- [String](#) [getName](#) () const  
*device info*
- [String](#) [getVendor](#) () const
- [String](#) [getVersion](#) () const
- const [Features](#) & **getFeatures** () const
- bool [hasQuery](#) (Query::Type type) const  
*device types*
- bool **hasShader** (Shader::Type type) const
- bool [hasTarget](#) (Format format) const  
*device formats*
- bool **hasTexture** (Format format) const
- bool **hasSurface** (Format format) const
- [Device](#) [createDevice](#) (uint32\_t index, uint32\_t frames=3) const  
*create device*
- [Device](#) **createCommandDevice** (uint32\_t frames=3) const
- [Device](#) **createComputeDevice** (uint32\_t frames=3) const
- [Device](#) **createCopyDevice** (uint32\_t frames=3) const
- [Query](#) [createQuery](#) () const  
*create query*
- [Query](#) **createQuery** (Query::Type type) const
- [Fence](#) [createFence](#) () const  
*create fence*
- [Fence](#) **createFence** ([Fence](#) &shared) const
- [Fence](#) **createFence** ([Fence::Flags](#) flags) const
- [Buffer](#) [createBuffer](#) () const  
*create buffer*
- [Buffer](#) **createBuffer** ([Buffer](#) &shared) const
- [Buffer](#) **createBuffer** ([Buffer::Flags](#) flags, size\_t size, Format format=FormatUnknown) const
- [Buffer](#) **createBuffer** ([Buffer::Flags](#) flags, const void \*src, size\_t size, Format format=FormatUnknown) const
- [Sampler](#) [createSampler](#) () const  
*create sampler*
- [Sampler](#) **createSampler** (const [Sampler](#) &sampler) const
- [Sampler](#) **createSampler** ([Sampler::Filter](#) filter, [Sampler::WrapMode](#) mode=Sampler::WrapModeRepeat, uint32\_t anisotropy=Sampler::MaxAnisotropy) const
- [Texture](#) [createTexture](#) () const  
*create texture*
- [Texture](#) **createTexture** ([Texture](#) &shared) const

- **Texture createTexture** (Texture::Type type, Format format, const [Size](#) &size, uint32\_t layers, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- **Texture createTexture** (Texture::Type type, Format format, const [Size](#) &size, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- **Texture createTexture** (const [Image](#) &image, [Texture::Flags](#) flags=Texture::DefaultFlags, [Async](#) \*async=nullptr) const
- **Texture createTexture2D** (Format format, uint32\_t size, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- **Texture createTexture3D** (Format format, uint32\_t size, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- **Texture createTextureCube** (Format format, uint32\_t size, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- **Texture createTexture2D** (Format format, uint32\_t width, uint32\_t height, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- **Texture createTexture3D** (Format format, uint32\_t width, uint32\_t height, uint32\_t depth, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- **Texture createTexture2D** (Format format, uint32\_t width, uint32\_t height, uint32\_t layers, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- **Texture createTextureCube** (Format format, uint32\_t size, uint32\_t layers, [Texture::Flags](#) flags=Texture::DefaultFlags) const
- **Texture loadTexture** (const char \*name, [Texture::Flags](#) flags=Texture::DefaultFlags, [Image::Flags](#) image\_flags=Image::FlagNone, uint32\_t offset=0, [Async](#) \*async=nullptr) const
- **Texture loadTexture** (const [String](#) &name, [Texture::Flags](#) flags=Texture::DefaultFlags, [Image::Flags](#) image\_flags=Image::FlagNone, uint32\_t offset=0, [Async](#) \*async=nullptr) const
- **Texture loadTexture** ([Stream](#) &stream, [Texture::Flags](#) flags=Texture::DefaultFlags, [Image::Flags](#) image\_flags=Image::FlagNone, uint32\_t offset=0, [Async](#) \*async=nullptr) const
- **Tracing createTracing** () const
  - create tracing*
- **Tracing createTracing** (const [Tracing](#) &tracing) const
- **Tracing createTracing** (uint32\_t num\_instances, [Buffer](#) instance\_buffer=Buffer::null, size\_t instance\_offset=0, [Tracing::Flags](#) flags=Tracing::DefaultFlags) const
- **Tracing createTracing** (uint32\_t num\_vertices, Format vertex\_format, size\_t vertex\_stride, uint32\_t num\_indices, Format index\_format, [Tracing::Flags](#) flags=Tracing::DefaultFlags) const
- **Tracing createTracing** (uint32\_t num\_bounds, size\_t bound\_stride, [Buffer](#) bound\_buffer=Buffer::null, size\_t bound\_offset=0, [Tracing::Flags](#) flags=Tracing::DefaultFlags) const
- **BufferTable createBufferTable** () const
  - create buffer table*
- **BufferTable createBufferTable** (uint32\_t size) const
- **BufferTable createBufferTable** (const Array< [Buffer](#) > &buffers, bool owner=false) const
- **TextureTable createTextureTable** () const
  - create texture table*
- **TextureTable createTextureTable** (Texture::Type type, uint32\_t size) const
- **TextureTable createTextureTable** (const Array< [Texture](#) > &textures, bool owner=false) const
- **Shader createShader** () const
  - create shader*
- **Shader loadShader** (Shader::Type type, const char \*name, const char \*format,...) const 1(4)
- **Shader Shader loadShaderGLSL** (Shader::Type type, const char \*name, const char \*format,...) const 1(4)
- **Shader Shader Shader loadShader** (Shader::Type type, const char \*name, const [String](#) &macros=[String::null](#)) const
- **Shader loadShaderGLSL** (Shader::Type type, const char \*name, const [String](#) &macros=[String::null](#)) const
- **Shader loadShaderSPIRV** (Shader::Type type, const char \*name) const
- **Shader createShader** (Shader::Type type, const char \*src, const char \*format,...) const 1(4)
- **Shader Shader createShaderGLSL** (Shader::Type type, const char \*src, const char \*format,...) const 1(4)
- **Shader Shader Shader createShader** (Shader::Type type, const char \*src, const [String](#) &macros=[String::null](#)) const
- **Shader createShaderGLSL** (Shader::Type type, const char \*src, const [String](#) &macros=[String::null](#)) const
- **Shader createShaderSPIRV** (Shader::Type type, const Array< uint32\_t > &data) const
- **Kernel createKernel** () const

- create kernel*
- **Kernel** **createKernel** (const **Kernel** &kernel) const
- void **releaseKernel** (**Kernel** &kernel) const
- release kernel*
- **Pipeline** **createPipeline** () const
- create pipeline*
- **Pipeline** **createPipeline** (const **Pipeline** &pipeline) const
- void **releasePipeline** (**Pipeline** &pipeline) const
- release pipeline*
- **Traversal** **createTraversal** () const
- create traversal*
- **Traversal** **createTraversal** (const **Traversal** &traversal) const
- void **releaseTraversal** (**Traversal** &traversal) const
- release traversal*
- **Target** **createTarget** () const
- create target*
- **Target** **createTarget** (**Surface** &surface) const
- **Target** **createTarget** (**Window** &window) const
- **Target** **createTarget** (const InitializerList< **Texture** > &textures, **Target::Operation** op=**Target::OpDefault**) const
- **Compute** **createCompute** () const
- create compute*
- **Command** **createCommand** () const
- create command*
- **Command** **createCommand** (**Target** &target) const
- bool **setBuffer** (**Buffer** &buffer, size\_t offset, const void \*src, size\_t size) const
- set buffer data*
- bool **setBuffer** (**Buffer** &buffer, const void \*src, size\_t size) const
- bool **setBuffer** (**Buffer** &buffer, const void \*src) const
- bool **getBuffer** (**Buffer** &buffer, size\_t offset, void \*dest, size\_t size) const
- get buffer data*
- bool **getBuffer** (**Buffer** &buffer, void \*dest, size\_t size) const
- bool **getBuffer** (**Buffer** &buffer, void \*dest) const
- void \* **mapBuffer** (**Buffer** &buffer, size\_t offset, size\_t size) const
- map buffer data*
- void \* **mapBuffer** (**Buffer** &buffer, size\_t size) const
- void \* **mapBuffer** (**Buffer** &buffer) const
- bool **unmapBuffer** (**Buffer** &buffer) const
- bool **copyBuffer** (**Buffer** &buffer, size\_t dest\_offset, **Buffer** &src, size\_t src\_offset, size\_t size) const
- copy buffer data*
- bool **copyBuffer** (**Buffer** &buffer, size\_t dest\_offset, **Buffer** &src, size\_t size) const
- bool **copyBuffer** (**Buffer** &buffer, **Buffer** &src, size\_t size) const
- bool **copyBuffer** (**Buffer** &buffer, **Buffer** &src) const
- bool **clearBuffer** (**Buffer** &buffer, Format format, size\_t offset, const void \*src, size\_t size) const
- clear buffer data*
- bool **clearBuffer** (**Buffer** &buffer, Format format, const void \*src, size\_t size) const
- bool **clearBuffer** (**Buffer** &buffer, Format format, const void \*src) const
- bool **clearBuffer** (**Buffer** &buffer) const
- bool **flushBuffer** (**Buffer** &buffer, **Buffer::Flags** flags=**Buffer::FlagNone**) const
- flush buffer data*
- bool **flushBuffers** (const Array< **Buffer** > &buffers, **Buffer::Flags** flags=**Buffer::FlagNone**) const
- bool **flushBuffers** (const InitializerList< **Buffer** > &buffers, **Buffer::Flags** flags=**Buffer::FlagNone**) const

- void **releaseBuffer** (Buffer &buffer) const  
*release buffer*
- void **releaseSampler** (Sampler &sampler) const  
*release sampler*
- bool **setTexture** (Texture &texture, const Origin &dest\_origin, const Slice &dest\_slice, const Image &image, const Slice &src\_slice) const  
*set texture data*
- bool **setTexture** (Texture &texture, const Origin &dest\_origin, const Image &image) const
- bool **setTexture** (Texture &texture, const Slice &dest\_slice, const Image &image) const
- bool **setTexture** (Texture &texture, const Image &image) const
- bool **getTexture** (Texture &texture, const Slice &src\_slice, Image &image, const Slice &dest\_slice) const  
*get texture data*
- bool **getTexture** (Texture &texture, Image &image, const Slice &dest\_slice) const
- bool **getTexture** (Texture &texture, Image &image) const
- bool **copyTexture** (Texture &texture, const Origin &dest\_origin, const Slice &dest\_slice, Texture &src, const Region &src\_region, const Slice &src\_slice) const  
*copy texture data*
- bool **copyTexture** (Texture &texture, const Origin &dest\_origin, Texture &src, const Region &src\_region) const
- bool **copyTexture** (Texture &texture, const Slice &dest\_slice, Texture &src, const Slice &src\_slice) const
- bool **copyTexture** (Texture &texture, Texture &src) const
- bool **clearTexture** (Texture &texture, const Region &region, const Slice &slice, const void \*src) const  
*clear texture data*
- bool **clearTexture** (Texture &texture, const Region &region, const void \*src) const
- bool **clearTexture** (Texture &texture, const Slice &slice, const void \*src) const
- bool **clearTexture** (Texture &texture, const void \*src) const
- bool **bindTexture** (Texture &texture, const Region \*regions, uint32\_t num\_regions, const Slice \*slices, uint32\_t num\_slices, bool commit, Fence &fence) const  
*bind texture memory*
- bool **bindTexture** (Texture &texture, const Region \*regions, uint32\_t num\_regions, const Slice \*slices, uint32\_t num\_slices, bool commit) const
- bool **bindTexture** (Texture &texture, const Region &region, const Slice &slice, bool commit, Fence &fence) const
- bool **bindTexture** (Texture &texture, const Region &region, const Slice &slice, bool commit) const
- bool **createMipmaps** (Texture &texture, const Slice &slice) const  
*create texture mipmaps*
- bool **createMipmaps** (Texture &texture) const
- bool **flushTexture** (Texture &texture, Texture::Flags flags=Texture::FlagNone) const  
*flush texture data*
- bool **flushTexture** (Texture &texture, const Slice &slice, Texture::Flags flags=Texture::FlagNone) const
- bool **flushTextures** (const Array< Texture > &textures, Texture::Flags flags=Texture::FlagNone) const
- bool **flushTextures** (const InitializerList< Texture > &textures, Texture::Flags flags=Texture::FlagNone) const
- void **releaseTexture** (Texture &texture) const  
*release texture*
- virtual bool **setTracing** (Tracing &tracing, const Tracing::Instance \*instances, uint32\_t num\_instances) const  
*set tracing data*
- bool **buildTracing** (Tracing &tracing, Buffer &buffer, Tracing::Flags flags=Tracing::FlagNone) const  
*build tracing*
- bool **buildTracing** (Tracing &tracing, Buffer &buffer, size\_t offset, Tracing::Flags flags=Tracing::FlagNone) const
- bool **buildTracings** (const Array< Tracing > &tracings, Buffer &buffer, Tracing::Flags flags=Tracing::FlagNone) const

- bool **buildTracings** (const Array< [Tracing](#) > &tracings, [Buffer](#) &buffer, size\_t offset, [Tracing::Flags](#) flags=[Tracing::FlagNone](#)) const
- bool **copyTracing** ([Tracing](#) &tracing, [Buffer](#) &buffer, size\_t offset=0) const  
*copy tracing address*
- bool **copyTracings** (const Array< [Tracing](#) > &tracings, [Buffer](#) &buffer, size\_t offset, size\_t stride=0) const
- bool **flushTracing** ([Tracing](#) &tracing) const  
*flush tracing*
- bool **flushTracings** (const Array< [Tracing](#) > &tracings) const
- void **releaseTracing** ([Tracing](#) &tracing) const  
*release tracing*
- bool **setBufferTable** ([BufferTable](#) &table, uint32\_t index, [Buffer](#) &buffer, bool owner=false) const  
*set table buffer*
- bool **setBufferTable** ([BufferTable](#) &table, uint32\_t index, const Array< [Buffer](#) > &buffers, bool owner=false) const
- void **releaseBufferTable** ([BufferTable](#) &table) const  
*release buffer table*
- bool **setTextureTable** ([TextureTable](#) &table, uint32\_t index, [Texture](#) &texture, bool owner=false) const  
*set table texture*
- bool **setTextureTable** ([TextureTable](#) &table, uint32\_t index, const Array< [Texture](#) > &textures, bool owner=false) const
- void **releaseTextureTable** ([TextureTable](#) &table) const  
*release texture table*
- bool **beginQuery** ([Query](#) &query) const  
*begin/end query*
- void **endQuery** ([Query](#) &query) const
- bool **copyQuery** ([Query](#) &query, [Buffer](#) &buffer, size\_t offset=0) const  
*copy query data*
- bool **copyQueries** (const Array< [Query](#) > &queries, [Buffer](#) &buffer, size\_t offset=0, size\_t stride=0) const
- bool **copyQueries** (const InitializerList< [Query](#) > &queries, [Buffer](#) &buffer, size\_t offset=0, size\_t stride=0) const
- bool **waitFence** ([Fence](#) &fence) const  
*fence synchronization*
- bool **signalFence** ([Fence](#) &fence) const
- bool **execute** ([Device](#) &device) const  
*execute context*
- bool **flip** ([Fence](#) &fence) const  
*flip context*
- bool **flip** () const
- bool **flush** () const  
*flush context*
- bool **finish** () const  
*finish context*
- bool **check** () const  
*check errors*

#### 5.91.1 Detailed Description

[Device](#) class

## 5.92 Tellusim::DialogColor Class Reference

```
#include <interface/TellusimDialogs.h>
```

### Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagAlpha** = (1 << 0),  
**FlagMouse** = (1 << 1),  
**DefaultFlags** = FlagNone,  
**NumFlags** = 2 }  
*dialog flags*
- enum [Result](#) {  
**ResultCancel** = 0,  
**ResultOk**,  
**NumResults** }  
*dialog result*
- using [ChangedCallback](#) = Function< void([Color](#))>  
*changed callback*
- using [UpdateCallback](#) = Function< bool()>  
*update callback*

### Public Member Functions

- **DialogColor** (const char \*title=nullptr, const [Color](#) &color=[Color::zero](#))
- **DialogColor** (const [String](#) &title, const [Color](#) &color=[Color::zero](#))
- void [setPosition](#) (int32\_t x, int32\_t y)  
*dialog position*
- int32\_t [getPositionX](#) () const
- int32\_t [getPositionY](#) () const
- void [setTitle](#) (const char \*title)  
*dialog title*
- void [setTitle](#) (const [String](#) &title)
- [String](#) [getTitle](#) () const
- void [setColor](#) (const [Color](#) &color, bool callback=false)  
*dialog color*
- const [Color](#) & [getColor](#) () const
- void [setChangedCallback](#) (const [ChangedCallback](#) &func)
- [ChangedCallback](#) [getChangedCallback](#) () const
- void [setUpdateCallback](#) (const [UpdateCallback](#) &func)
- [UpdateCallback](#) [getUpdateCallback](#) () const
- [Result](#) [run](#) ([Flags](#) flags=DefaultFlags)  
*run dialog*

#### 5.92.1 Detailed Description

### [DialogColor](#)

## 5.93 Tellusim::DialogDirectory Class Reference

```
#include <interface/TellusimDialogs.h>
```

## Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagMouse** = (1 << 0),  
**DefaultFlags** = FlagNone,  
**NumFlags** = 1 }  
*dialog flags*
- enum [Result](#) {  
**ResultCancel** = 0,  
**ResultOk**,  
**NumResults** }  
*dialog result*
- using [UpdateCallback](#) = Function< bool()>  
*update callback*

## Public Member Functions

- **DialogDirectory** (const char \*title=nullptr, const char \*name=nullptr)
- **DialogDirectory** (const [String](#) &title, const char \*name=nullptr)
- **DialogDirectory** (const char \*title, const [String](#) &name)
- **DialogDirectory** (const [String](#) &title, const [String](#) &name)
- void [setPosition](#) (int32\_t x, int32\_t y)  
*dialog position*
- int32\_t [getPositionX](#) () const
- int32\_t [getPositionY](#) () const
- void [setTitle](#) (const char \*title)  
*dialog title*
- void [setTitle](#) (const [String](#) &title)
- [String](#) [getTitle](#) () const
- void [setDirectory](#) (const char \*name)  
*dialog directory*
- void [setDirectory](#) (const [String](#) &name)
- [String](#) [getDirectory](#) () const
- void [setUpdateCallback](#) (const [UpdateCallback](#) &func)
- [UpdateCallback](#) [getUpdateCallback](#) () const
- [Result](#) [run](#) ([Flags](#) flags=DefaultFlags)  
*run dialog*

## 5.93.1 Detailed Description

[DialogDirectory](#)

## 5.94 Tellusim::DialogFileOpen Class Reference

```
#include <interface/TellusimDialogs.h>
```

## Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagHidden** = (1 << 0),  
**FlagMouse** = (1 << 1),  
**DefaultFlags** = FlagNone,  
**NumFlags** = 2 }  
*dialog flags*
- enum [Result](#) {  
**ResultCancel** = 0,  
**ResultOk**,  
**NumResults** }  
*dialog result*
- using [UpdateCallback](#) = Function< bool()>  
*update callback*

## Public Member Functions

- **DialogFileOpen** (const char \*title=nullptr, const char \*name=nullptr)
- **DialogFileOpen** (const [String](#) &title, const char \*name=nullptr)
- **DialogFileOpen** (const char \*title, const [String](#) &name)
- **DialogFileOpen** (const [String](#) &title, const [String](#) &name)
- void [setPosition](#) (int32\_t x, int32\_t y)  
*dialog position*
- int32\_t **getPositionX** () const
- int32\_t **getPositionY** () const
- void [setTitle](#) (const char \*title)  
*dialog title*
- void **setTitle** (const [String](#) &title)
- [String](#) **getTitle** () const
- void [setFilter](#) (const char \*filter)  
*dialog filter*
- void **setFilter** (const [String](#) &filter)
- [String](#) **getFilter** () const
- void [setFile](#) (const char \*name)  
*dialog file*
- void **setFile** (const [String](#) &name)
- [String](#) **getFile** () const
- void **setUpdateCallback** (const [UpdateCallback](#) &func)
- [UpdateCallback](#) **getUpdateCallback** () const
- [Result](#) **run** ([Flags](#) flags=DefaultFlags)  
*run dialog*

## 5.94.1 Detailed Description

[DialogFileOpen](#)

## 5.95 Tellusim::DialogFileSave Class Reference

```
#include <interface/TellusimDialogs.h>
```



## Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagHidden** = (1 << 0),  
**FlagOverwrite** = (1 << 1),  
**FlagMouse** = (1 << 2),  
**DefaultFlags** = (FlagOverwrite),  
**NumFlags** = 3 }  
*dialog flags*
- enum [Result](#) {  
**ResultCancel** = 0,  
**ResultOk**,  
**NumResults** }  
*dialog result*
- using [UpdateCallback](#) = Function< bool()>  
*update callback*

## Public Member Functions

- **DialogFileSave** (const char \*title=nullptr, const char \*name=nullptr)
- **DialogFileSave** (const [String](#) &title, const char \*name=nullptr)
- **DialogFileSave** (const char \*title, const [String](#) &name)
- **DialogFileSave** (const [String](#) &title, const [String](#) &name)
- void [setPosition](#) (int32\_t x, int32\_t y)  
*dialog position*
- int32\_t [getPositionX](#) () const
- int32\_t [getPositionY](#) () const
- void [setTitle](#) (const char \*title)  
*dialog title*
- void [setTitle](#) (const [String](#) &title)
- [String](#) [getTitle](#) () const
- void [setFilter](#) (const char \*filter)  
*dialog filter*
- void [setFilter](#) (const [String](#) &filter)
- [String](#) [getFilter](#) () const
- void [setFile](#) (const char \*name)  
*dialog file*
- void [setFile](#) (const [String](#) &name)
- [String](#) [getFile](#) () const
- void [setUpdateCallback](#) (const [UpdateCallback](#) &func)
- [UpdateCallback](#) [getUpdateCallback](#) () const
- [Result](#) [run](#) ([Flags](#) flags=DefaultFlags)  
*run dialog*

## 5.95.1 Detailed Description

[DialogFileSave](#)

## 5.96 Tellusim::DialogMenu Class Reference

```
#include <interface/TellusimDialogs.h>
```

## Public Types

- enum **Flags** {  
**FlagNone** = 0,  
**FlagMouse** = (1 << 0),  
**DefaultFlags** = FlagNone,  
**NumFlags** = 1 }  
*dialog flags*
- enum **Result** {  
**ResultCancel** = 0,  
**ResultClick**,  
**NumResults** }  
*dialog result*
- using **ClickedCallback** = Function< void()>  
*click item*
- using **ChangedCallback** = Function< void(bool)>  
*check item*
- using **UpdateCallback** = Function< bool()>  
*update callback*

## Public Member Functions

- void **setPosition** (int32\_t x, int32\_t y)  
*dialog position*
- int32\_t **getPositionX** () const
- int32\_t **getPositionY** () const
- uint32\_t **getNumItems** () const  
*number of items*
- void **setItemText** (uint32\_t index, const char \*text)  
*item text*
- void **setItemText** (uint32\_t index, const **String** &text)
- **String** **getItemText** (uint32\_t index) const
- void **setItemKey** (uint32\_t index, const char \*key)  
*item key*
- **String** **getItemKey** (uint32\_t index) const
- void **setItemImage** (uint32\_t index, const **Image** &image)  
*item image*
- **Image** **getItemImage** (uint32\_t index) const
- void **setItemChecked** (uint32\_t index, bool checked, bool callback=false)  
*item checked*
- bool **isItemChecked** (uint32\_t index) const
- void **setItemEnabled** (uint32\_t index, bool enabled)  
*item enabled*
- bool **isItemEnabled** (uint32\_t index) const
- void **setItemHidden** (uint32\_t index, bool hidden)  
*item hidden*
- bool **isItemHidden** (uint32\_t index) const
- void **setItemsGroup** (uint32\_t index, uint32\_t size)  
*item group*
- uint32\_t **getItemGroupIndex** (uint32\_t index) const
- uint32\_t **getItemGroupSize** (uint32\_t index) const
- uint32\_t **addItem** (const char \*text, const char \*key=nullptr)

*text item*

- uint32\_t **addItem** (const [String](#) &text, const char \*key=nullptr)
- uint32\_t **addItem** (const char \*text, const [Image](#) &image, const char \*key=nullptr)
- uint32\_t **addItem** (const [String](#) &text, const [Image](#) &image, const char \*key=nullptr)
- uint32\_t **addItem** (const char \*text, const [ClickedCallback](#) &func, const char \*key=nullptr)
- uint32\_t **addItem** (const [String](#) &text, const [ClickedCallback](#) &func, const char \*key=nullptr)
- uint32\_t **addItem** (const char \*text, const [Image](#) &image, const [ClickedCallback](#) &func, const char \*key=nullptr)
- uint32\_t **addItem** (const [String](#) &text, const [Image](#) &image, const [ClickedCallback](#) &func, const char \*key=nullptr)
- [ClickedCallback](#) **getItemClickedCallback** (uint32\_t index) const
- uint32\_t **addItem** (const char \*text, bool checked, const [ChangedCallback](#) &func, const char \*key=nullptr)
- uint32\_t **addItem** (const [String](#) &text, bool checked, const [ChangedCallback](#) &func, const char \*key=nullptr)
- uint32\_t **addItem** (const char \*text, const [Image](#) &image, bool checked, const [ChangedCallback](#) &func, const char \*key=nullptr)
- uint32\_t **addItem** (const [String](#) &text, const [Image](#) &image, bool checked, const [ChangedCallback](#) &func, const char \*key=nullptr)
- [ChangedCallback](#) **getItemChangedCallback** (uint32\_t index) const
- void **setUpdateCallback** (const [UpdateCallback](#) &func)
- [UpdateCallback](#) **getUpdateCallback** () const
- [Result](#) **run** ([Flags](#) flags=DefaultFlags)

*run dialog*

#### 5.96.1 Detailed Description

[DialogMenu](#)

## 5.97 Tellusim::DialogMessage Class Reference

```
#include <interface/TellusimDialogs.h>
```

### Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagYes** = (1 << 0),  
**FlagNo** = (1 << 1),  
**FlagOk** = (1 << 2),  
**FlagCancel** = (1 << 3),  
**FlagClose** = (1 << 4),  
**FlagMessage** = (1 << 5),  
**FlagWarning** = (1 << 6),  
**FlagQuestion** = (1 << 7),  
**FlagError** = (1 << 8),  
**FlagMouse** = (1 << 9),  
**FlagYesNo** = (FlagYes | FlagNo),  
**FlagOkCancel** = (FlagOk | FlagCancel),  
**DefaultFlags** = (FlagOk),  
**NumFlags** = 10 }

*dialog flags*

- enum [Result](#) {  
**ResultClose** = 0,  
**ResultCancel**,  
**ResultOk**,  
**ResultNo**,  
**ResultYes**,  
**NumResults** }  
*dialog result*
- using [UpdateCallback](#) = Function< bool()>  
*update callback*

#### Public Member Functions

- **DialogMessage** (const char \*title=NULLptr, const char \*message=NULLptr)
- **DialogMessage** (const [String](#) &title, const char \*message=NULLptr)
- **DialogMessage** (const char \*title, const [String](#) &message)
- **DialogMessage** (const [String](#) &title, const [String](#) &message)
- void [setPosition](#) (int32\_t x, int32\_t y)  
*dialog position*
- int32\_t [getPositionX](#) () const
- int32\_t [getPositionY](#) () const
- void [setTitle](#) (const char \*title)  
*dialog title*
- void [setTitle](#) (const [String](#) &title)
- [String](#) [getTitle](#) () const
- void [setMessage](#) (const char \*message)  
*dialog message*
- void [setMessage](#) (const [String](#) &message)
- [String](#) [getMessage](#) () const
- void [setUpdateCallback](#) (const [UpdateCallback](#) &func)
- [UpdateCallback](#) [getUpdateCallback](#) () const
- [Result](#) [run](#) ([Flags](#) flags=DefaultFlags)  
*run dialog*

#### 5.97.1 Detailed Description

#### [DialogMessage](#)

### 5.98 Tellusim::DialogProgress Class Reference

```
#include <interface/TellusimDialogs.h>
```

#### Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagMouse** = (1 << 0),  
**DefaultFlags** = FlagNone,  
**NumFlags** = 1 }  
*dialog flags*
- enum [Result](#) {  
**ResultCancel** = 0,  
**ResultOk**,  
**NumResults** }  
*dialog result*

## Public Member Functions

- **DialogProgress** (const char \*title=nullptr, const char \*message=nullptr)
- **DialogProgress** (const [String](#) &title, const char \*message=nullptr)
- **DialogProgress** (const char \*title, const [String](#) &message)
- **DialogProgress** (const [String](#) &title, const [String](#) &message)
- void **setPosition** (int32\_t x, int32\_t y)  
*dialog position*
- int32\_t **getPositionX** () const
- int32\_t **getPositionY** () const
- void **setTitle** (const char \*title)  
*dialog title*
- void **setTitle** (const [String](#) &title)
- [String](#) **getTitle** () const
- void **setMessage** (const char \*message)  
*dialog message*
- void **setMessage** (const [String](#) &message)
- [String](#) **getMessage** () const
- void **setProgress** (uint32\_t progress)  
*dialog progress*
- uint32\_t **getProgress** () const
- [Result](#) **run** ([Flags](#) flags=DefaultFlags)  
*run dialog*
- void **close** ()  
*close dialog*

## 5.98.1 Detailed Description

[DialogProgress](#)

## 5.99 Tellusim::Directory Class Reference

```
#include <core/TellusimDirectory.h>
```

## Public Types

- enum [Attributes](#) {  
**AttributeNone** = 0,  
**AttributeRead** = (1 << 0),  
**AttributeWrite** = (1 << 1),  
**AttributeHidden** = (1 << 2),  
**AttributeExecute** = (1 << 3),  
**AttributeTemporary** = (1 << 4),  
**NumAttributes** = 5 }  
*attributes*

## Public Member Functions

- bool **open** (const char \*name, bool children=false)  
*open/close directory*
- bool **open** (const **String** &name, bool children=false)
- void **close** ()
- bool **isOpened** () const  
*directory status*
- **String** **getName** () const
- uint32\_t **getNumFiles** () const  
*files*
- **String** **getFileName** (uint32\_t index) const
- **Attributes** **getFileAttributes** (uint32\_t index) const
- uint64\_t **getFileMTime** (uint32\_t index) const
- uint64\_t **getFileATime** (uint32\_t index) const
- uint64\_t **getFileCTime** (uint32\_t index) const
- size\_t **getFileSize** (uint32\_t index) const
- const Array< **String** > **getFiles** () const
- uint32\_t **getNumDirectories** () const  
*directories*
- **String** **getDirectoryName** (uint32\_t index) const
- **Attributes** **getDirectoryAttributes** (uint32\_t index) const
- uint64\_t **getDirectoryCTime** (uint32\_t index) const
- uint32\_t **getDirectorySize** (uint32\_t index) const
- const Array< **String** > **getDirectories** () const

## Static Public Member Functions

- static bool **isFile** (const char \*name)  
*file utils*
- static bool **isFile** (const **String** &name)
- static bool **setFileAttributes** (const char \*name, **Attributes** attributes)
- static **Attributes** **getFileAttributes** (const char \*name)
- static bool **setFileMTime** (const char \*name, uint64\_t time)
- static uint64\_t **getFileMTime** (const char \*name)
- static uint64\_t **getFileATime** (const char \*name)
- static uint64\_t **getFileCTime** (const char \*name)
- static size\_t **getFileSize** (const char \*name)
- static size\_t **getFileSize** (const **String** &name)
- static bool **removeFile** (const char \*name)
- static bool **removeFile** (const **String** &name)
- static bool **copyFile** (const char \*name, const char \*new\_name, bool attributes=false)
- static bool **copyFile** (const **String** &name, const **String** &new\_name, bool attributes=false)
- static bool **isDirectory** (const char \*name)  
*directory utils*
- static bool **isDirectory** (const **String** &name)
- static bool **changeDirectory** (const char \*name)
- static bool **changeDirectory** (const **String** &name)
- static bool **createDirectory** (const char \*name, bool children=false)
- static bool **createDirectory** (const **String** &name, bool children=false)
- static bool **removeDirectory** (const char \*name, bool children=false)
- static bool **removeDirectory** (const **String** &name, bool children=false)
- static bool **copyDirectory** (const char \*name, const char \*new\_name, bool attributes=false)

- static bool **copyDirectory** (const [String](#) &name, const [String](#) &new\_name, bool attributes=false)
- static bool **rename** (const char \*name, const char \*new\_name)
- static bool **rename** (const [String](#) &name, const [String](#) &new\_name)
- static [String](#) **getCurrentDirectory** ()  
*common directories*
- static [String](#) **getHomeDirectory** ()
- static [String](#) **getTempDirectory** ()
- static [String](#) **getConfigDirectory** ()
- static [String](#) **getDocumentsDirectory** ()
- static const Array< [String](#) > **getDriveNames** ()  
*drive utils*

#### 5.99.1 Detailed Description

[Directory](#) class

### 5.100 Tellusim::Compute::DispatchIndirect Struct Reference

dispatch indirect parameters

```
#include <platform/TellusimCompute.h>
```

#### Public Attributes

- uint32\_t **group\_width**
- uint32\_t **group\_height**
- uint32\_t **group\_depth**
- uint32\_t **padding**

#### 5.100.1 Detailed Description

dispatch indirect parameters

### 5.101 Tellusim::BitonicSort::DispatchParameters Struct Reference

#### Public Attributes

- uint32\_t **keys\_offset**
- uint32\_t **data\_offset**
- uint32\_t **size**
- uint32\_t **padding**

### 5.102 Tellusim::PrefixScan::DispatchParameters Struct Reference

#### Public Attributes

- uint32\_t **offset**
- uint32\_t **size**
- uint32\_t **padding\_0**
- uint32\_t **padding\_1**

### 5.103 Tellusim::RadixSort::DispatchParameters Struct Reference

#### Public Attributes

- uint32\_t **keys\_offset**
- uint32\_t **data\_offset**
- uint32\_t **size**
- uint32\_t **padding**

### 5.104 Tellusim::SpatialGrid::DispatchParameters Struct Reference

#### Public Attributes

- uint32\_t **offset**
- uint32\_t **size**
- uint32\_t **padding\_0**
- uint32\_t **padding\_1**

### 5.105 Tellusim::SpatialTree::DispatchParameters Struct Reference

#### Public Attributes

- uint32\_t **offset**
- uint32\_t **size**
- uint32\_t **padding\_0**
- uint32\_t **padding\_1**

### 5.106 Tellusim::Command::DrawArraysIndirect Struct Reference

draw arrays indirect parameters

```
#include <platform/TellusimCommand.h>
```

#### Public Attributes

- uint32\_t **num\_vertices**
- uint32\_t **num\_instances**
- uint32\_t **base\_vertex**
- uint32\_t **base\_instance**

#### 5.106.1 Detailed Description

draw arrays indirect parameters



## 5.107 Tellusim::Command::DrawElementsIndirect Struct Reference

draw elements indirect parameters

```
#include <platform/TellusimCommand.h>
```

### Public Attributes

- uint32\_t **num\_indices**
- uint32\_t **num\_instances**
- uint32\_t **base\_index**
- int32\_t **base\_vertex**
- uint32\_t **base\_instance**

#### 5.107.1 Detailed Description

draw elements indirect parameters

## 5.108 Tellusim::Command::DrawMeshIndirect Struct Reference

draw mesh indirect parameters

```
#include <platform/TellusimCommand.h>
```

### Public Attributes

- uint32\_t **group\_width**
- uint32\_t **group\_height**
- uint32\_t **group\_depth**
- uint32\_t **padding**

#### 5.108.1 Detailed Description

draw mesh indirect parameters

## 5.109 Tellusim::EncoderASTC Class Reference

```
#include <graphics/TellusimEncoderASTC.h>
```

## Public Types

- enum [Mode](#) {  
**ModeASTC44RGBAu8n** = 0,  
**ModeASTC54RGBAu8n**,  
**ModeASTC55RGBAu8n**,  
**NumModes** }  
*Encoder modes.*
- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagASTC44RGBAu8n** = (1 << ModeASTC44RGBAu8n),  
**FlagASTC54RGBAu8n** = (1 << ModeASTC54RGBAu8n),  
**FlagASTC55RGBAu8n** = (1 << ModeASTC55RGBAu8n),  
**FlagCube** = (1 << (NumModes + 1)),  
**FlagsAll** = (FlagASTC44RGBAu8n | FlagASTC54RGBAu8n | FlagASTC55RGBAu8n) }  
*Encoder flags.*

## Public Member Functions

- void [clear](#) ()  
*clear encoder*
- bool [isCreated](#) ([Mode](#) mode) const  
*check encoder*
- bool [create](#) (const [Device](#) &device, [Mode](#) mode)  
*create encoder*
- bool [create](#) (const [Device](#) &device, [Flags](#) flags)
- bool [dispatch](#) ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src, const [Slice](#) &dest\_slice, const [Slice](#) &src\_slice, uint32\_t components=4) const
- bool [dispatch](#) ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src, const [Slice](#) &src\_slice, uint32\_t components=4) const
- bool [dispatch](#) ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src, uint32\_t components=4) const

## 5.109.1 Detailed Description

[EncoderASTC](#) class

## 5.109.2 Member Function Documentation

5.109.2.1 [dispatch\(\)](#)

```
bool Tellusim::EncoderASTC::dispatch (
    Compute & compute,
    Mode mode,
    Texture & dest,
    Texture & src,
    const Slice & dest_slice,
    const Slice & src_slice,
    uint32_t components = 4 ) const
```

[dispatch](#) texture encoder

## Parameters

|                   |                                             |
|-------------------|---------------------------------------------|
| <i>mode</i>       | Compression mode.                           |
| <i>dest</i>       | Destination proxy texture.                  |
| <i>src</i>        | <a href="#">Source</a> texture to compress. |
| <i>dest_slice</i> | Destination texture slice.                  |
| <i>src_slice</i>  | <a href="#">Source</a> texture slice.       |
| <i>components</i> | Number of components.                       |

## 5.110 Tellusim::EncoderBC15 Class Reference

```
#include <graphics/TellusimEncoderBC15.h>
```

## Public Types

- enum [Mode](#) {  
**ModeBC1RGBu8n** = 0,  
**ModeBC2RGBAu8n**,  
**ModeBC3RGBAu8n**,  
**ModeBC4Ru8n**,  
**ModeBC5RGu8n**,  
**NumModes** }  
*Encoder modes.*
- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagBC1RGBu8n** = (1 << ModeBC1RGBu8n),  
**FlagBC2RGBAu8n** = (1 << ModeBC2RGBAu8n),  
**FlagBC3RGBAu8n** = (1 << ModeBC3RGBAu8n),  
**FlagBC4Ru8n** = (1 << ModeBC4Ru8n),  
**FlagBC5RGu8n** = (1 << ModeBC5RGu8n),  
**FlagCube** = (1 << (NumModes + 1)),  
**FlagsBC13** = (FlagBC1RGBu8n | FlagBC2RGBAu8n | FlagBC3RGBAu8n),  
**FlagsBC45** = (FlagBC4Ru8n | FlagBC5RGu8n),  
**FlagsAll** = (FlagsBC13 | FlagsBC45) }  
*Encoder flags.*

## Public Member Functions

- void [clear](#) ()  
*clear encoder*
- bool [isCreated](#) ([Mode](#) mode) const  
*check encoder*
- bool [create](#) (const [Device](#) &device, [Mode](#) mode)  
*create encoder*
- bool [create](#) (const [Device](#) &device, [Flags](#) flags)
- bool [dispatch](#) ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src, const [Slice](#) &dest\_slice, const [Slice](#) &src\_slice) const
- bool [dispatch](#) ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src, const [Slice](#) &src\_slice) const
- bool [dispatch](#) ([Compute](#) &compute, [Mode](#) mode, [Texture](#) &dest, [Texture](#) &src) const

### 5.110.1 Detailed Description

[EncoderBC15](#) class

### 5.110.2 Member Function Documentation

#### 5.110.2.1 dispatch()

```
bool Tellusim::EncoderBC15::dispatch (
    Compute & compute,
    Mode mode,
    Texture & dest,
    Texture & src,
    const Slice & dest_slice,
    const Slice & src_slice ) const
```

dispatch texture encoder

#### Parameters

|                   |                                             |
|-------------------|---------------------------------------------|
| <i>mode</i>       | Compression mode.                           |
| <i>dest</i>       | Destination proxy texture.                  |
| <i>src</i>        | <a href="#">Source</a> texture to compress. |
| <i>dest_slice</i> | Destination texture slice.                  |
| <i>src_slice</i>  | <a href="#">Source</a> texture slice.       |

## 5.111 Tellusim::EncoderBC67 Class Reference

```
#include <graphics/TellusimEncoderBC67.h>
```

#### Public Types

- enum [Mode](#) {  
**ModeBC6RGBf16s** = 0,  
**ModeBC6RGBf16u**,  
**ModeBC7RGBAu8n**,  
**NumModes** }  
*Encoder modes.*
- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagBC6RGBf16s** = (1 << ModeBC6RGBf16s),  
**FlagBC6RGBf16u** = (1 << ModeBC6RGBf16u),  
**FlagBC7RGBAu8n** = (1 << ModeBC7RGBAu8n),  
**FlagCube** = (1 << (NumModes + 1)),  
**FlagsAll** = (FlagBC6RGBf16s | FlagBC6RGBf16u | FlagBC7RGBAu8n) }  
*Encoder flags.*

## Public Member Functions

- void **clear** ()  
*clear encoder*
- bool **isCreated** (Mode mode) const  
*check encoder*
- bool **create** (const Device &device, Mode mode)  
*create encoder*
- bool **create** (const Device &device, Flags flags)
- bool **dispatch** (Compute &compute, Mode mode, Texture &dest, Texture &src, const Slice &dest\_slice, const Slice &src\_slice, uint32\_t components=4) const
- bool **dispatch** (Compute &compute, Mode mode, Texture &dest, Texture &src, const Slice &src\_slice, uint32\_t components=4) const
- bool **dispatch** (Compute &compute, Mode mode, Texture &dest, Texture &src, uint32\_t components=4) const

## 5.111.1 Detailed Description

EncoderBC67 class

## 5.111.2 Member Function Documentation

## 5.111.2.1 dispatch()

```
bool Tellusim::EncoderBC67::dispatch (
    Compute & compute,
    Mode mode,
    Texture & dest,
    Texture & src,
    const Slice & dest_slice,
    const Slice & src_slice,
    uint32_t components = 4 ) const
```

dispatch texture encoder

## Parameters

|                   |                             |
|-------------------|-----------------------------|
| <i>mode</i>       | Compression mode.           |
| <i>dest</i>       | Destination proxy texture.  |
| <i>src</i>        | Source texture to compress. |
| <i>dest_slice</i> | Destination texture slice.  |
| <i>src_slice</i>  | Source texture slice.       |
| <i>components</i> | Number of components.       |

## 5.112 Tellusim::f32u32 Union Reference

```
#include <math/TellusimFloat.h>
```

**Public Member Functions**

- **f32u32** (float32\_t f)
- **f32u32** (uint32\_t u)
- **f32u32** (int32\_t i)

**Public Attributes**

- float32\_t **f**
- uint32\_t **u**
- int32\_t **i**

**5.112.1 Detailed Description**

32-bit floating-point as integer

**5.113 Tellusim::f64u64 Union Reference**

```
#include <math/TellusimFloat.h>
```

**Public Member Functions**

- **f64u64** (float64\_t f)
- **f64u64** (uint64\_t u)
- **f64u64** (int64\_t i)

**Public Attributes**

- float64\_t **f**
- uint64\_t **u**
- int64\_t **i**

**5.113.1 Detailed Description**

64-bit floating-point as integer

**5.114 Tellusim::Face Struct Reference**

```
#include <TellusimTypes.h>
```

**Public Member Functions**

- **Face** (uint32\_t base)
- **Face** (uint32\_t base, uint32\_t size)

## Public Attributes

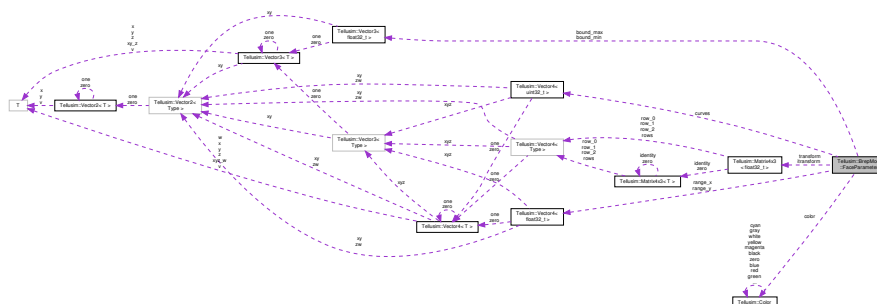
- uint32\_t **base** = 0
- uint32\_t **size** = 1

### 5.114.1 Detailed Description

Slice

### 5.115 Tellusim::BrepModel::FaceParameters Struct Reference

Collaboration diagram for Tellusim::BrepModel::FaceParameters:



## Public Attributes

- **Matrix4x3f transform**
- **Matrix4x3f itransform**
- **Vector3f bound\_min**
- **float32\_t brep\_winding**
- **Vector3f bound\_max**
- **float32\_t face\_winding**
- **Color color**
- **Vector4u curves**
- **Vector4f range\_x**
- **Vector4f range\_y**
- 

```
union {
    struct {
        float32_t radius
    } sphere
    struct {
        float32_t radius
    } cylinder
    struct {
        float32_t radius
        float32_t angle
    } cone
    struct {
        float32_t radius_0
```

```

        float32_t radius_1
    } torus
    struct {
        float32_t radius_0
        float32_t radius_1
    } lemon
    struct {
        float32_t height
        float32_t pcoord_y
        uint32_t vertex_index
        uint32_t padding
        float32_t length [2]
    } revolve
    struct {
        float32_t width
        float32_t pcoord_x
        uint32_t vertex_index
        uint32_t padding
        float32_t direction [3]
    } extrude
    struct {
        float32_t width
        float32_t height
        float32_t pcoord_x
        float32_t pcoord_y
        uint32_t vertex_index
        uint32_t padding [3]
        float32_t length [4]
    } surface
    struct {
        float32_t parameters_0 [4]
        float32_t parameters_1 [4]
        float32_t parameters_2 [4]
    }
};

```

### 5.116 Tellusim::Device::Features Struct Reference

device features

```
#include <platform/TellusimDevice.h>
```

#### Public Attributes

- bool **threadAccess**
- bool **sparseBuffer**
- bool **bufferTable**
- bool **sparseTexture**
- bool **sparseArrayTexture**
- bool **cubeArrayTexture**
- bool **textureTable**
- bool **baseInstanceIndex**
- bool **drawIndirectIndex**
- bool **drawIndirectCount**



- bool **taskIndirectCount**
- bool **vertexStorage**
- bool **vertexIndexLayer**
- bool **geometryPassthrough**
- bool **fragmentBarycentric**
- bool **fragmentStencilExport**
- bool **dualSourceBlending**
- bool **depthRangeOneToOne**
- bool **conservativeRaster**
- bool **conditionalRendering**
- bool **rayTracing**
- bool **computeTracing**
- bool **fragmentTracing**
- bool **indirectTracing**
- uint32\_t **recursionDepth**
- bool **subgroupVote**
- bool **subgroupMath**
- bool **subgroupShuffle**
- uint32\_t **subgroupSize**
- uint32\_t **minSubgroupSize**
- uint32\_t **maxSubgroupSize**
- bool **shaderu8**
- bool **shaderf16**
- bool **shaderu16**
- bool **shaderf64**
- bool **shaderu64**
- bool **atomicGroupf32**
- bool **atomicGroupu64**
- bool **atomicBufferf32**
- bool **atomicBufferu64**
- bool **atomicTexturef32**
- bool **atomicTextureu32**
- bool **atomicTextureu64**
- bool **matrix16f16**
- bool **matrix16x8x8f16**
- bool **matrix16x8x16f16**
- bool **matrix16f16f32**
- bool **matrix16x8x8f16f32**
- bool **matrix16x8x16f16f32**
- uint32\_t **uniformAlignment**
- uint32\_t **storageAlignment**
- uint32\_t **maxTextureSamples**
- uint32\_t **maxTexture2DSize**
- uint32\_t **maxTexture3DSize**
- uint32\_t **maxTextureLayers**
- uint32\_t **maxGroupSizeX**
- uint32\_t **maxGroupSizeY**
- uint32\_t **maxGroupSizeZ**
- uint32\_t **maxGroupCountX**
- uint32\_t **maxGroupCountY**
- uint32\_t **maxGroupCountZ**
- uint32\_t **maxTaskCount**
- uint32\_t **maxTaskMemory**
- uint32\_t **maxTaskMeshes**
- uint32\_t **maxMeshMemory**

- uint32\_t **maxMeshVertices**
- uint32\_t **maxMeshPrimitives**
- uint32\_t **maxViewportCount**
- uint32\_t **maxClipCullCount**
- uint64\_t **maxUniformSize**
- uint64\_t **maxStorageSize**
- uint32\_t **groupMemory**
- uint64\_t **videoMemory**
- uint32\_t **vendorID**
- uint32\_t **deviceId**
- uint32\_t **pciBusID**
- uint32\_t **pciDomainID**
- uint32\_t **pciDeviceID**

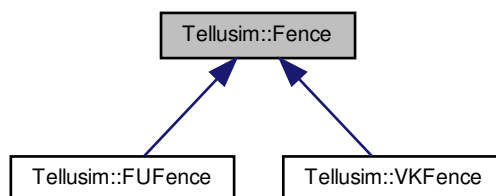
#### 5.116.1 Detailed Description

device features

### 5.117 Tellusim::Fence Class Reference

```
#include <platform/TellusimFence.h>
```

Inheritance diagram for Tellusim::Fence:



#### Public Types

- enum **Flags** {  
**FlagNone** = 0,  
**FlagSemaphore** = (1 << 0),  
**FlagSignaled** = (1 << 1),  
**FlagShared** = (1 << 2),  
**FlagExtern** = (1 << 3),  
**DefaultFlags** = FlagNone,  
**NumFlags** = 4 }  
*Flags* flags.

## Public Member Functions

- Platform [getPlatform](#) () const  
*fence platform*
- const char \* [getPlatformName](#) () const
- uint32\_t [getIndex](#) () const  
*fence device index*
- void [clear](#) ()  
*clear fence*
- bool [isCreated](#) () const  
*check fence*
- bool [create](#) (Flags flags=DefaultFlags)  
*create fence*
- Flags [getFlags](#) () const  
*fence flags*
- bool [hasFlag](#) (Flags flags) const
- bool [hasFlags](#) (Flags flags) const
- String [getFlagsName](#) () const
- String [getDescription](#) () const  
*fence description*

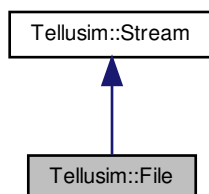
## 5.117.1 Detailed Description

[Fence](#) class

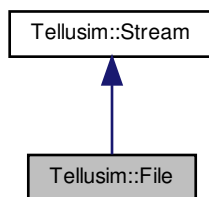
## 5.118 Tellusim::File Class Reference

```
#include <core/TellusimFile.h>
```

Inheritance diagram for Tellusim::File:



Collaboration diagram for Tellusim::File:



### Public Member Functions

- bool **open** (const char \*name, const char \*mode)  
*open/close file*
- bool **open** (const [String](#) &name, const char \*mode)
- bool **open** (int32\_t fd, const char \*name, const char \*mode)
- bool **popen** (const char \*command, const char \*mode)
- bool **popen** (const [String](#) &command, const char \*mode)
- void **close** ()

### Static Public Member Functions

- static bool **isFile** (const char \*name)  
*file utils*
- static bool **isFile** (const [String](#) &name)
- static uint64\_t **getMTime** (const char \*name)
- static size\_t **getSize** (const char \*name)
- static bool **remove** (const char \*name)

#### 5.118.1 Detailed Description

[File](#) class

### 5.119 Tellusim::FixedPool< Type, Index > Class Template Reference

```
#include <core/TellusimPool.h>
```

## Public Member Functions

- **FixedPool** (uint32\_t size)  
*initialize pool*
- void **init** (uint32\_t size)  
*initialize pool*
- void **shutdown** ()  
*shutdown pool*
- bool **isInitialized** ()  
*pool status*
- uint32\_t **allocate** ()  
*allocate object memory*
- void **free** (uint32\_t offset)  
*free object memory*
- uint32\_t **getMemory** () const  
*memory usage in bytes*
- uint32\_t **getAllocations** () const  
*number of allocations*

## 5.119.1 Detailed Description

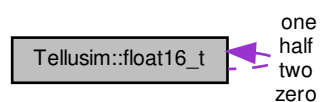
```
template<class Type, class Index = uint32_t>
class Tellusim::FixedPool< Type, Index >
```

Fixed pool memory allocator

## 5.120 Tellusim::float16\_t Struct Reference

```
#include <math/TellusimFloat.h>
```

Collaboration diagram for Tellusim::float16\_t:



## Public Member Functions

- **float16\_t** (uint16\_t u)
- **float16\_t** (float32\_t f)
- **float16\_t** (float64\_t f)
- void **set** (float32\_t f)  
*handle normal, denormal, infinity and NaN cases*
- float32\_t **get** () const
- void **setFast** (float32\_t f)  
*handle normal cases only*
- float32\_t **getFast** () const
- **operator uint16\_t** () const  
*conversion to numbers*
- **operator float32\_t** () const
- **operator float64\_t** () const
- **float16\_t** & **operator=** (uint16\_t u)  
*conversion from numbers*
- **float16\_t** & **operator=** (float32\_t f)
- **float16\_t** & **operator=** (float16\_t f)
- uint16\_t **exponent** () const  
*access to number*
- uint16\_t **mantissa** () const

## Public Attributes

- uint16\_t **bits**

## Static Public Attributes

- static const **float16\_t** **zero**  
*constant values*
- static const **float16\_t** **half**
- static const **float16\_t** **one**
- static const **float16\_t** **two**

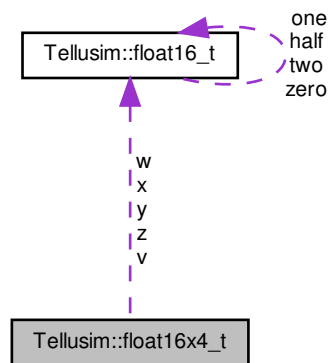
## 5.120.1 Detailed Description

16-bit floating-point number class mantissa bits 10, exponent bits 5 strict conversion is performed by default

## 5.121 Tellusim::float16x4\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

Collaboration diagram for Tellusim::float16x4\_t:



## Public Types

- enum { **Size** = 4 }

## Public Member Functions

- [float16x4\\_t](#) (const [float32x4\\_t](#) &v)
- [float16x4\\_t](#) (const [float16\\_t](#) \*v)
- [float16x4\\_t](#) (const [float16\\_t](#) \*v, [float16\\_t](#) w)
- [float16x4\\_t](#) ([float16\\_t](#) v)
- [float16x4\\_t](#) ([float16\\_t](#) x, [float16\\_t](#) y, [float16\\_t](#) z, [float16\\_t](#) w)
- [float16x4\\_t](#) (uint64\_t v)
- [float16x4\\_t](#) (const [float16x4\\_t](#) &v)
- void [set](#) ([float16x4\\_t](#) &v)  
*update vector data*
- void [set](#) ([float16\\_t](#) X, [float16\\_t](#) Y, [float16\\_t](#) Z, [float16\\_t](#) W)
- void [set](#) (const [float16\\_t](#) \*1 v, [float16\\_t](#) W)
- void [set](#) (const [float16\\_t](#) \*1 v)
- void [get](#) ([float16\\_t](#) \*1 v) const
- template<uint32\_t Index>  
void [set](#) ([float16\\_t](#) V)
- template<uint32\_t Index>  
[float16\\_t](#) [get](#) () const
- [float16x4\\_t](#) & [operator=](#) (const [float16x4\\_t](#) &v)  
*assignment operator*

## Public Attributes

```

•
union {
    struct {
        float16_t x
        float16_t y
        float16_t z
        float16_t w
    }
    float16_t v [Size]
    uint64_t vec
};

```

## 5.121.1 Detailed Description

Vector of four `float16_t` components

## 5.121.2 Constructor &amp; Destructor Documentation

5.121.2.1 `float16x4_t()`

```

Tellusim::float16x4_t::float16x4_t (
    const float32x4_t & v ) [explicit]

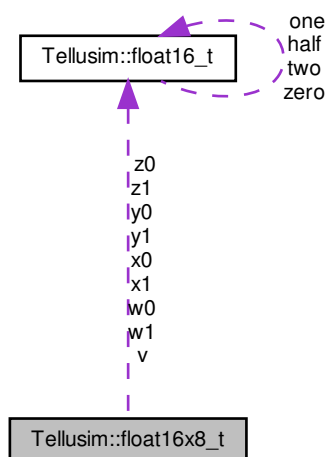
```

Vector of four `float16_t` components

## 5.122 Tellusim::float16x8\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

Collaboration diagram for Tellusim::float16x8\_t:





## Public Types

- enum { **Size** = 8 }

## Public Member Functions

- [float16x8\\_t](#) (const [float32x8\\_t](#) &v)
- [float16x8\\_t](#) (const [float16\\_t](#) \*v)
- [float16x8\\_t](#) (const uint64\_t \*v)
- [float16x8\\_t](#) (uint64\_t v0, uint64\_t v1)
- [float16x8\\_t](#) ([float16\\_t](#) v)
- [float16x8\\_t](#) (const [float16x4\\_t](#) &v0, const [float16x4\\_t](#) &v1)
- [float16x8\\_t](#) ([float16\\_t](#) x0, [float16\\_t](#) y0, [float16\\_t](#) z0, [float16\\_t](#) w0, [float16\\_t](#) x1, [float16\\_t](#) y1, [float16\\_t](#) z1, [float16\\_t](#) w1)
- [int16x8\\_t](#) [asi16x8](#) () const  
*cast vector data*
- [uint16x8\\_t](#) [asu16x8](#) () const
- [int32x4\\_t](#) [asi32x4](#) () const
- [uint32x4\\_t](#) [asu32x4](#) () const
- [float32x4\\_t](#) [asf32x4](#) () const  
*cast vector data*
- void [set](#) (const [float16x8\\_t](#) &v)  
*update vector data*
- void [set](#) ([float16\\_t](#) X0, [float16\\_t](#) Y0, [float16\\_t](#) Z0, [float16\\_t](#) W0, [float16\\_t](#) X1, [float16\\_t](#) Y1, [float16\\_t](#) Z1, [float16\\_t](#) W1)
- void [set](#) (const [float16\\_t](#) \*1 v)
- void [get](#) ([float16\\_t](#) \*1 v) const
- void [set](#) (const uint64\_t \*1 v)
- void [get](#) (uint64\_t \*1 v) const
- template<uint32\_t Index>  
void [set](#) ([float16\\_t](#) V)
- template<uint32\_t Index>  
[float16\\_t](#) [get](#) () const
- [float16x8\\_t](#) & [operator=](#) (const [float16x8\\_t](#) &v)  
*assignment operator*

## Public Attributes

```

•
union {
    struct {
        float16\_t x0
        float16\_t y0
        float16\_t z0
        float16\_t w0
        float16\_t x1
        float16\_t y1
        float16\_t z1
        float16\_t w1
    }
    struct {
        uint64_t vec0
        uint64_t vec1
    }
    float16\_t v [Size]
    uint64_t vec [2]
};

```

### 5.122.1 Detailed Description

Vector of eight `float16_t` components

### 5.122.2 Constructor & Destructor Documentation

#### 5.122.2.1 `float16x8_t()`

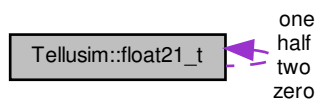
```
Tellusim::float16x8_t::float16x8_t (
    const float32x8_t & v ) [explicit]
```

Vector of eight `float16_t` components

## 5.123 Tellusim::float21\_t Struct Reference

```
#include <math/TellusimFloat.h>
```

Collaboration diagram for Tellusim::float21\_t:



### Public Member Functions

- `float21_t` (`uint32_t` u)
- `float21_t` (`float32_t` f)
- `float21_t` (`float64_t` f)
- void `set` (`float32_t` f)
  - handle normal, denormal, infinity and NaN cases*
- `float32_t` `get` () const
- void `setFast` (`float32_t` f)
  - handle normal cases only*
- `float32_t` `getFast` () const
- `operator uint32_t` () const
  - conversion to numbers*
- `operator float32_t` () const
- `operator float64_t` () const
- `float21_t` & `operator=` (`uint32_t` u)
  - conversion from numbers*
- `float21_t` & `operator=` (`float32_t` f)
- `float21_t` & `operator=` (`float21_t` f)
- `uint32_t` `exponent` () const
  - access to number*
- `uint32_t` `mantissa` () const

## Public Attributes

- uint32\_t **bits**

## Static Public Attributes

- static const float21\_t **zero**  
*constant values*
- static const float21\_t **half**
- static const float21\_t **one**
- static const float21\_t **two**

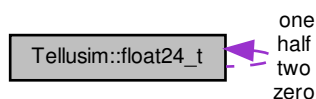
## 5.123.1 Detailed Description

21-bit floating-point number class mantissa bits 14, exponent bits 6 strict conversion is performed by default

## 5.124 Tellusim::float24\_t Struct Reference

```
#include <math/TellusimFloat.h>
```

Collaboration diagram for Tellusim::float24\_t:



## Public Member Functions

- **float24\_t** (uint32\_t u)
- **float24\_t** (float32\_t f)
- **float24\_t** (float64\_t f)
- void **set** (float32\_t f)  
*handle normal, denormal, infinity and NaN cases*
- float32\_t **get** () const
- void **setFast** (float32\_t f)  
*handle normal cases only*
- float32\_t **getFast** () const
- **operator uint32\_t** () const  
*conversion to numbers*
- **operator float32\_t** () const
- **operator float64\_t** () const
- **float24\_t** & **operator=** (uint32\_t u)  
*conversion from numbers*
- **float24\_t** & **operator=** (float32\_t f)
- **float24\_t** & **operator=** (float24\_t f)
- uint32\_t **exponent** () const  
*access to number*
- uint32\_t **mantissa** () const

## Public Attributes

- uint32\_t **bits**

## Static Public Attributes

- static const float24\_t **zero**  
*constant values*
- static const float24\_t **half**
- static const float24\_t **one**
- static const float24\_t **two**

## 5.124.1 Detailed Description

24-bit floating-point number class mantissa bits 17, exponent bits 6 strict conversion is performed by default

## 5.125 Tellusim::float32x4\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

## Public Types

- enum { **Size** = 4 }

## Public Member Functions

- float32x4\_t (const float64x4\_t &v)
- float32x4\_t (const float32\_t \*v)
- float32x4\_t (const float32\_t \*v, float32\_t w)
- float32x4\_t (float32\_t v)
- float32x4\_t (const int32x4\_t &v)
- float32x4\_t (const uint32x4\_t &v)
- float32x4\_t (const float16x4\_t &v)
- float32x4\_t (float32\_t x, float32\_t y, float32\_t z, float32\_t w=0.0f)
- int32x4\_t asi32x4 () const  
*cast vector data*
- uint32x4\_t asu32x4 () const
- int16x8\_t asi16x8 () const
- uint16x8\_t asu16x8 () const
- float16x8\_t asf16x8 () const
- void set (const float32x4\_t &v)  
*update vector data*
- void set (float32\_t X, float32\_t Y, float32\_t Z, float32\_t W)
- void set (const float32\_t \*1 v, float32\_t W)
- void set (const float32\_t \*1 v)
- void get (float32\_t \*1 v) const
- template<uint32\_t Index>  
void set (float32\_t V)

- `template<uint32_t Index>`  
`float32_t` **get** () const
- `template<uint32_t Index>`  
`float32x4_t` **get4** () const
- `float32x4_t` & **operator\*=** (float32\_t v)  
*vector to scalar operators*
- `float32x4_t` & **operator/=** (float32\_t v)
- `float32x4_t` & **operator+=** (float32\_t v)
- `float32x4_t` & **operator-=** (float32\_t v)
- `float32x4_t` & **operator\*=** (const float32x4\_t &v)  
*vector to vector operators*
- `float32x4_t` & **operator/=** (const float32x4\_t &v)
- `float32x4_t` & **operator+=** (const float32x4\_t &v)
- `float32x4_t` & **operator-=** (const float32x4\_t &v)
- `float32x4_t` **zxyw** () const  
*swizzle vector*
- `float32x4_t` **zwny** () const
- `float32x4_t` **yxwz** () const
- `float32_t` **sum** () const  
*sum vector components*

#### Public Attributes

- union {  
  struct {  
    float32\_t x  
    float32\_t y  
    float32\_t z  
    float32\_t w  
  }  
  float32\_t v [Size]  
};

#### 5.125.1 Detailed Description

Vector of four float32\_t components

#### 5.125.2 Constructor & Destructor Documentation

##### 5.125.2.1 float32x4\_t()

```
Tellusim::float32x4_t::float32x4_t (  
    const float64x4_t & v ) [explicit]
```

Vector of four float32\_t components

## 5.126 Tellusim::float32x8\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

### Public Types

- enum { **Size** = 8 }

### Public Member Functions

- [float32x8\\_t](#) (const [float64x8\\_t](#) &v)
- **float32x8\_t** (const float32\_t \*v)
- **float32x8\_t** (float32\_t v)
- **float32x8\_t** (const [int32x8\\_t](#) &v)
- **float32x8\_t** (const [uint32x8\\_t](#) &v)
- **float32x8\_t** (const [float32x4\\_t](#) &v0, const [float32x4\\_t](#) &v1)
- **float32x8\_t** (const [float16x8\\_t](#) &v)
- **float32x8\_t** (float32\_t x0, float32\_t y0, float32\_t z0, float32\_t w0, float32\_t x1, float32\_t y1, float32\_t z1, float32\_t w1)
- [int32x8\\_t](#) **asi32x8** () const  
*cast vector data*
- [uint32x8\\_t](#) **asu32x8** () const
- void **set** (const [float32x8\\_t](#) &v)  
*update vector data*
- void **set** (float32\_t X0, float32\_t Y0, float32\_t Z0, float32\_t W0, float32\_t X1, float32\_t Y1, float32\_t Z1, float32\_t W1)
- void **set** (const float32\_t \*1 v)
- void **get** (float32\_t \*1 v) const
- template<uint32\_t Index>  
void **set** (float32\_t V)
- template<uint32\_t Index>  
float32\_t **get** () const
- template<uint32\_t Index>  
[float32x8\\_t](#) **get8** () const
- [float32x8\\_t](#) & **operator\*=** (float32\_t v)  
*vector to scalar operators*
- [float32x8\\_t](#) & **operator/=** (float32\_t v)
- [float32x8\\_t](#) & **operator+=** (float32\_t v)
- [float32x8\\_t](#) & **operator-=** (float32\_t v)
- [float32x8\\_t](#) & **operator\*=** (const [float32x8\\_t](#) &v)  
*vector to vector operators*
- [float32x8\\_t](#) & **operator/=** (const [float32x8\\_t](#) &v)
- [float32x8\\_t](#) & **operator+=** (const [float32x8\\_t](#) &v)
- [float32x8\\_t](#) & **operator-=** (const [float32x8\\_t](#) &v)
- [float32x8\\_t](#) **xyzw10** () const  
*swizzle vector*
- [float32x8\\_t](#) **zwxxy01** () const
- [float32x8\\_t](#) **yxwz01** () const
- [float32x4\\_t](#) **xyzw0** () const
- [float32x4\\_t](#) **xyzw1** () const
- float32\_t **sum** () const  
*sum vector components*

## Public Attributes

- ```
union {
    struct {
        float32_t x0
        float32_t y0
        float32_t z0
        float32_t w0
        float32_t x1
        float32_t y1
        float32_t z1
        float32_t w1
    }
    float32_t v [Size]
};
```

## 5.126.1 Detailed Description

Vector of eight float32\_t components

## 5.126.2 Constructor &amp; Destructor Documentation

## 5.126.2.1 float32x8\_t()

```
Tellusim::float32x8_t::float32x8_t (
    const float64x8_t & v ) [explicit]
```

Vector of eight float32\_t components

## 5.127 Tellusim::float64x2\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

## Public Types

- enum { **Size** = 2 }

## Public Member Functions

- **float64x2\_t** (const float64\_t \*v)
- **float64x2\_t** (float64\_t v)
- **float64x2\_t** (float64\_t x, float64\_t y)
- void **set** (const float64x2\_t &v)  
*update vector data*
- void **set** (float64\_t X, float64\_t Y)
- void **set** (const float64\_t \*1 v)
- void **get** (float64\_t \*1 v) const
- template<uint32\_t Index>  
void **set** (float64\_t V)
- template<uint32\_t Index>  
float64\_t **get** () const
- template<uint64\_t Index>  
float64x2\_t **get2** () const
- float64x2\_t & **operator\*=** (float64\_t v)  
*vector to scalar operators*
- float64x2\_t & **operator/=** (float64\_t v)
- float64x2\_t & **operator+=** (float64\_t v)
- float64x2\_t & **operator-=** (float64\_t v)
- float64x2\_t & **operator\*=** (const float64x2\_t &v)  
*vector to vector operators*
- float64x2\_t & **operator/=** (const float64x2\_t &v)
- float64x2\_t & **operator+=** (const float64x2\_t &v)
- float64x2\_t & **operator-=** (const float64x2\_t &v)
- float64x2\_t **yx** () const  
*swizzle vector*
- float64\_t **sum** () const  
*sum vector components*

## Public Attributes

- union {  
  struct {  
    float64\_t x  
    float64\_t y  
  }  
  float64\_t v [Size]  
};

## 5.127.1 Detailed Description

Vector of two float64\_t components

## 5.128 Tellusim::float64x4\_t Struct Reference

```
#include <math/TellusimSimd.h>
```



## Public Types

- enum { **Size** = 4 }

## Public Member Functions

- **float64x4\_t** (const float64\_t \*v)
- **float64x4\_t** (const float64\_t \*v, float64\_t w)
- **float64x4\_t** (float64\_t v)
- **float64x4\_t** (const int32x4\_t &v)
- **float64x4\_t** (const uint32x4\_t &v)
- **float64x4\_t** (const float32x4\_t &v)
- **float64x4\_t** (const float64x2\_t &v0, const float64x2\_t &v1)
- **float64x4\_t** (float64\_t x, float64\_t y, float64\_t z, float64\_t w=0.0)
- void **set** (const float64x4\_t &v)

*update vector data*

- void **set** (float64\_t X, float64\_t Y, float64\_t Z, float64\_t W)
- void **set** (const float64\_t \*1 v, float64\_t W)
- void **set** (const float64\_t \*1 v)
- void **get** (float64\_t \*1 v) const
- template<uint32\_t Index>  
void **set** (float64\_t V)
- template<uint32\_t Index>  
float64\_t **get** () const
- template<uint64\_t Index>  
float64x4\_t **get4** () const
- float64x4\_t & **operator\*=** (float64\_t v)

*vector to scalar operators*

- float64x4\_t & **operator/=** (float64\_t v)
- float64x4\_t & **operator+=** (float64\_t v)
- float64x4\_t & **operator-=** (float64\_t v)
- float64x4\_t & **operator\*=** (const float64x4\_t &v)

*vector to vector operators*

- float64x4\_t & **operator/=** (const float64x4\_t &v)
- float64x4\_t & **operator+=** (const float64x4\_t &v)
- float64x4\_t & **operator-=** (const float64x4\_t &v)
- float64x4\_t **xywz** () const

*swizzle vector*

- float64x4\_t **zwxxy** () const
- float64x4\_t **yxwz** () const
- float64x2\_t **xy** () const
- float64x2\_t **zw** () const
- float64\_t **sum** () const

*sum vector components*

## Public Attributes

- union {  
  struct {  
    float64\_t **x**  
    float64\_t **y**  
    float64\_t **z**  
    float64\_t **w**  
  }  
  float64\_t **v** [**Size**]  
};

### 5.128.1 Detailed Description

Vector of four float64\_t components

## 5.129 Tellusim::float64x8\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

### Public Types

- enum { **Size** = 8 }

### Public Member Functions

- **float64x8\_t** (const float64\_t \*v)
- **float64x8\_t** (float64\_t v)
- **float64x8\_t** (const [int32x8\\_t](#) &v)
- **float64x8\_t** (const [uint32x8\\_t](#) &v)
- **float64x8\_t** (const [float32x8\\_t](#) &v)
- **float64x8\_t** (const [float64x4\\_t](#) &v0, const [float64x4\\_t](#) &v1)
- **float64x8\_t** (float64\_t x0, float64\_t y0, float64\_t z0, float64\_t w0, float64\_t x1, float64\_t y1, float64\_t z1, float64\_t w1)
- void **set** (const [float64x8\\_t](#) &v)  
*update vector data*
- void **set** (float64\_t X0, float64\_t Y0, float64\_t Z0, float64\_t W0, float64\_t X1, float64\_t Y1, float64\_t Z1, float64\_t W1)
- void **set** (const float64\_t \*1 v)
- void **get** (float64\_t \*1 v) const
- template<uint32\_t Index>  
void **set** (float64\_t V)
- template<uint32\_t Index>  
[float64\\_t](#) **get** () const
- template<uint64\_t Index>  
[float64x8\\_t](#) **get8** () const
- [float64x8\\_t](#) & **operator\*=** (float64\_t v)  
*vector to scalar operators*
- [float64x8\\_t](#) & **operator/=** (float64\_t v)
- [float64x8\\_t](#) & **operator+=** (float64\_t v)
- [float64x8\\_t](#) & **operator-=** (float64\_t v)
- [float64x8\\_t](#) & **operator\*=** (const [float64x8\\_t](#) &v)  
*vector to vector operators*
- [float64x8\\_t](#) & **operator/=** (const [float64x8\\_t](#) &v)
- [float64x8\\_t](#) & **operator+=** (const [float64x8\\_t](#) &v)
- [float64x8\\_t](#) & **operator-=** (const [float64x8\\_t](#) &v)
- [float64x8\\_t](#) **xyzw10** () const  
*swizzle vector*
- [float64x8\\_t](#) **zwxxy01** () const
- [float64x8\\_t](#) **yxwz01** () const
- [float64x4\\_t](#) **xyzw0** () const
- [float64x4\\_t](#) **xyzw1** () const
- float64\_t **sum** () const  
*sum vector components*

## Public Attributes

- ```

union {
    struct {
        float64_t x0
        float64_t y0
        float64_t z0
        float64_t w0
        float64_t x1
        float64_t y1
        float64_t z1
        float64_t w1
    }
    float64_t v [Size]
};

```

## 5.129.1 Detailed Description

Vector of eight float64\_t components

## 5.130 Tellusim::Font Class Reference

```
#include <interface/TellusimFont.h>
```

## Public Member Functions

- void **clear** ()  
*clear font*
- bool **isLoading** () const  
*check font*
- bool **load** (const char \*name)  
*load font*
- bool **load** (Stream &stream)
- float32\_t **getAdvance** (const FontStyle &style, uint32\_t code)  
*glyph advance*
- Rect **getRect** (const Vector3f &position, const FontStyle &style, const char \*str)  
*text rectangle*
- Rect **getRect** (const Vector3f &position, const FontStyle &style, const uint32\_t \*str)
- template<class Type >  
Rect **getRect** (const Vector3f &position, uint32\_t size, const Type \*str)
- Rect **getRect** (const FontBatch \*batches, uint32\_t num\_batches)  
*batched text rectangle*
- Rect **getRect** (const FontBatch32 \*batches, uint32\_t num\_batches)
- void **create** (const Device &device, const FontStyle &style, const char \*str)  
*create text*
- void **create** (const Device &device, const FontStyle &style, const uint32\_t \*str)
- template<class Type >  
void **create** (const Device &device, uint32\_t size, const Type \*str)
- void **create** (const Device &device, const FontBatch \*batches, uint32\_t num\_batches)

*create batched text*

- void **create** (const [Device](#) &device, const [FontBatch32](#) \*batches, uint32\_t num\_batches)
- void **draw** ([Command](#) &command, const [Vector3f](#) &position, const [FontStyle](#) &style, const char \*str)

*draw text*

- void **draw** ([Command](#) &command, const [Vector3f](#) &position, const [FontStyle](#) &style, const uint32\_t \*str)
- template<class Type >  
void **draw** ([Command](#) \*command, const [Vector3f](#) &position, uint32\_t size, const Type \*str)
- template<class Type >  
void **draw** ([Command](#) \*command, const [Vector3f](#) &position, uint32\_t size, const [Color](#) &color, const Type \*str)
- void **draw** ([Command](#) &command, const [FontBatch](#) \*batches, uint32\_t num\_batches)

*draw batched text*

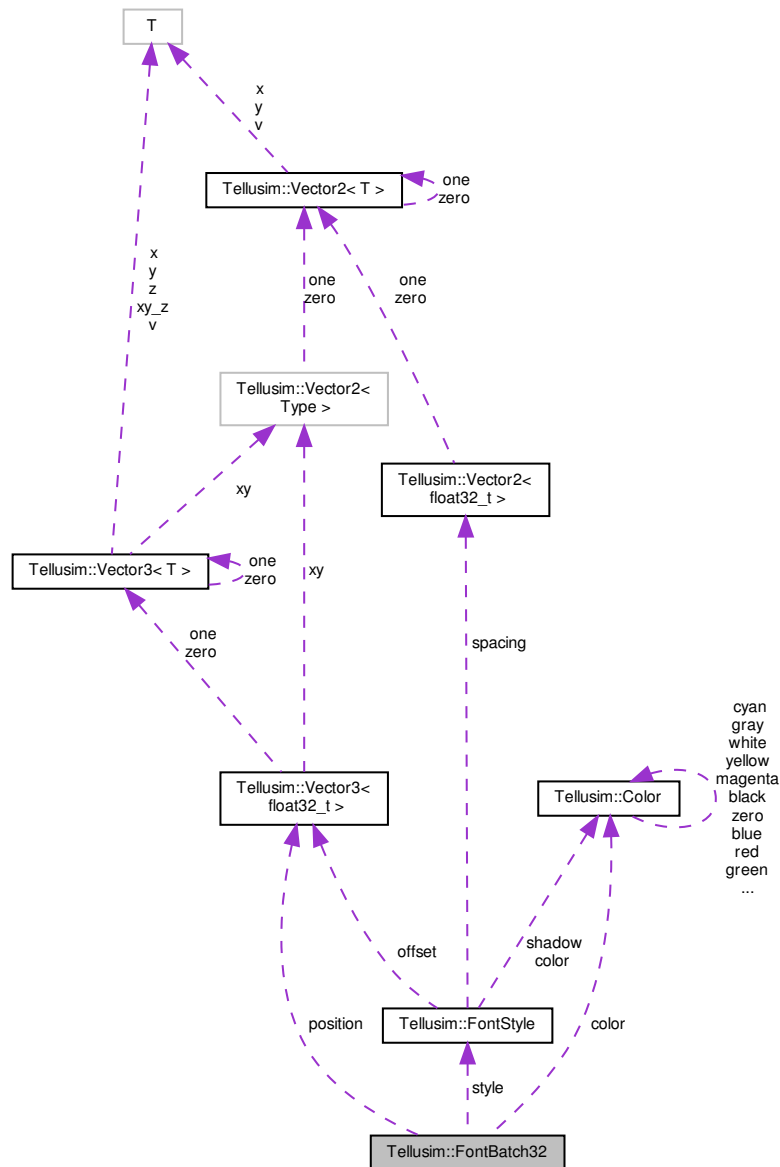
- void **draw** ([Command](#) &command, const [FontBatch32](#) \*batches, uint32\_t num\_batches)
- bool **flush** (const [Device](#) &device)

*flush textures***5.130.1 Detailed Description**[Font](#)



### 5.132 Tellusim::FontBatch32 Struct Reference

Collaboration diagram for Tellusim::FontBatch32:



#### Public Member Functions

- **FontBatch32** (const [Vector3f](#) &position, const uint32\_t \*str)
- **FontBatch32** (const [Vector3f](#) &position, const [FontStyle](#) \*style, const uint32\_t \*str)

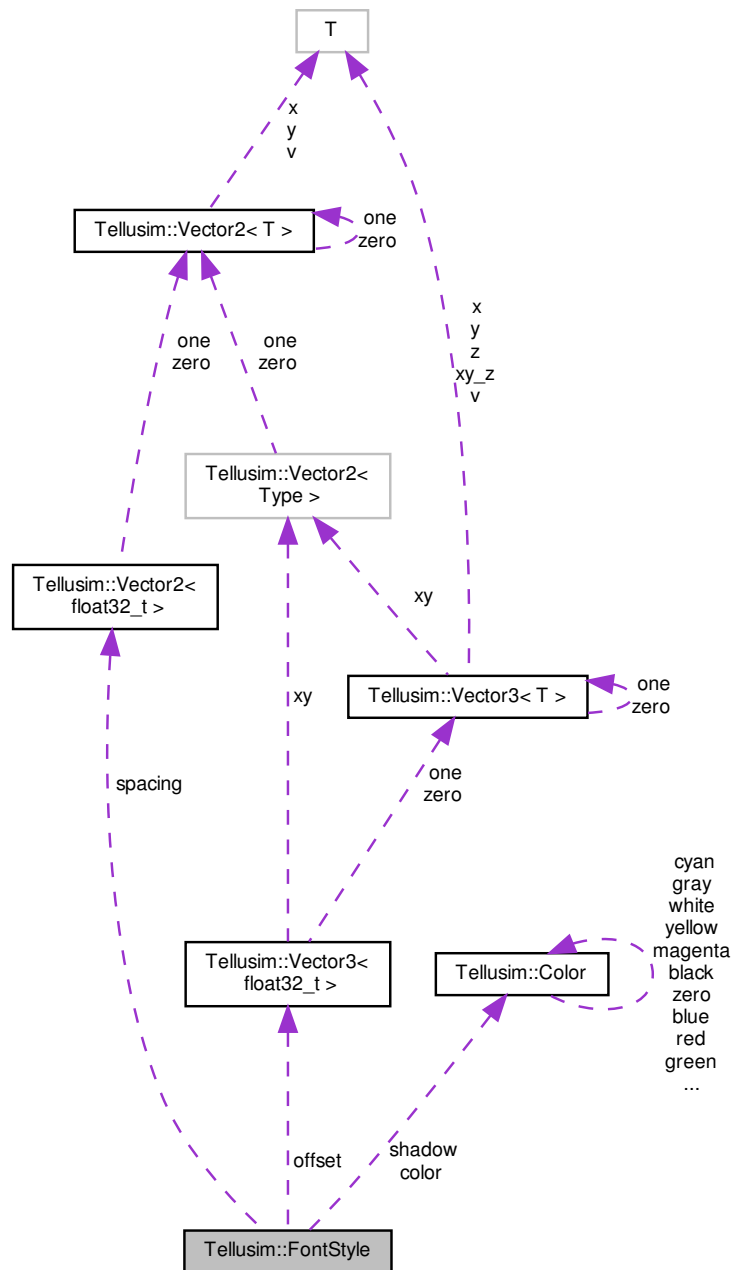
#### Public Attributes

- [Vector3f](#) **position** = [Vector3f::zero](#)
- [Color](#) **color** = [Color::white](#)
- const [FontStyle](#) \* **style** = nullptr
- const uint32\_t \* **str** = nullptr

## 5.133 Tellusim::FontStyle Struct Reference

```
#include <interface/TellusimTypes.h>
```

Collaboration diagram for Tellusim::FontStyle:



## Public Member Functions

- **FontStyle** (uint32\_t size)
- **FontStyle** (const [Color](#) &color)
- **FontStyle** (uint32\_t size, const [Color](#) &color)

## Public Attributes

- uint32\_t **size** = 16
- uint32\_t **scale** = 100
- bool **fixed** = false
- bool **kerning** = true
- [Vector2f](#) **spacing** = [Vector2f::zero](#)
- [Color](#) **color** = [Color::white](#)
- [Vector3f](#) **offset** = [Vector3f::zero](#)
- [Color](#) **shadow** = [Color::black](#)

## 5.133.1 Detailed Description

[FontStyle](#)

## 5.134 Tellusim::FourierTransform Class Reference

```
#include <parallel/TellusimFourierTransform.h>
```

## Public Types

- enum [Mode](#) {  
**ModeRf16i** = 0,  
**ModeRf32i**,  
**ModeRGf16i**,  
**ModeRGf32i**,  
**ModeRGBf16c**,  
**ModeRGBf21c**,  
**ModeRGBf16p**,  
**ModeRGBf32p**,  
**NumModes** }  
*Transform modes.*
- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagRf16i** = (1 << ModeRf16i),  
**FlagRf32i** = (1 << ModeRf32i),  
**FlagRGf16i** = (1 << ModeRGf16i),  
**FlagRGf32i** = (1 << ModeRGf32i),  
**FlagRGBf16c** = (1 << ModeRGBf16c),  
**FlagRGBf21c** = (1 << ModeRGBf21c),  
**FlagRGBf16p** = (1 << ModeRGBf16p),  
**FlagRGBf32p** = (1 << ModeRGBf32p),  
**FlagsInterleaved** = (FlagRf16i | FlagRf32i | FlagRGf16i | FlagRGf32i),  
**FlagsComplex** = (FlagRGBf16c | FlagRGBf21c),  
**FlagsPlanar** = (FlagRGBf16p | FlagRGBf32p),  
**FlagsAll** = (FlagsInterleaved | FlagsComplex | FlagsPlanar) }  
*Transform flags.*
- enum [Operation](#) {  
**ForwardCtoC** = 0,  
**BackwardCtoC**,  
**ForwardRtoC**,  
**BackwardCtoR**,  
**NumOperations** }  
*Transform operations.*



## Public Member Functions

- void **clear** ()  
*clear transform*
- bool **isCreated** (**Mode** mode) const  
*check transform*
- bool **isCreated** (**Flags** flags) const
- uint32\_t **getMaxWidth** () const  
*transform parameters*
- uint32\_t **getMaxHeight** () const
- bool **create** (const **Device** &device, **Mode** mode, uint32\_t width, uint32\_t height, **Async** \*async=nullptr)  
*create transform*
- bool **create** (const **Device** &device, **Flags** flags, uint32\_t width, uint32\_t height, **Async** \*async=nullptr)
- bool **dispatch** (**Compute** &compute, **Mode** mode, **Operation** op, **Texture** &dest, **Texture** &src, const **Slice** &dest\_slice, const **Slice** &src\_slice) const
- bool **dispatch** (**Compute** &compute, **Mode** mode, **Operation** op, **Texture** &dest, **Texture** &src, const **Slice** &src\_slice) const
- bool **dispatch** (**Compute** &compute, **Mode** mode, **Operation** op, **Texture** &dest, **Texture** &src) const

## 5.134.1 Detailed Description

**FourierTransform** class

## 5.134.2 Member Function Documentation

## 5.134.2.1 dispatch()

```
bool Tellusim::FourierTransform::dispatch (
    Compute & compute,
    Mode mode,
    Operation op,
    Texture & dest,
    Texture & src,
    const Slice & dest_slice,
    const Slice & src_slice ) const
```

dispatch transform

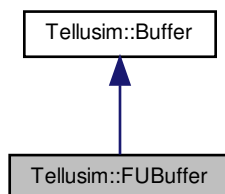
## Parameters

|             |                        |
|-------------|------------------------|
| <i>mode</i> | Transform mode.        |
| <i>op</i>   | Transform operation.   |
| <i>dest</i> | Destination texture.   |
| <i>src</i>  | <b>Source</b> texture. |

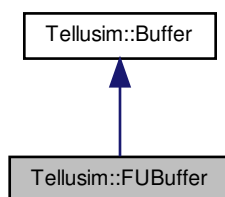
### 5.135 Tellusim::FUBuffer Class Reference

```
#include <platform/TellusimBuffer.h>
```

Inheritance diagram for Tellusim::FUBuffer:



Collaboration diagram for Tellusim::FUBuffer:



#### Public Member Functions

- **FUBuffer** (const Array< [Buffer](#) > &buffers, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumBuffers](#) () const  
*Fusion buffers.*
- const [Buffer](#) **getBuffer** (uint32\_t index) const
- [Buffer](#) **getBuffer** (uint32\_t index)

#### Additional Inherited Members

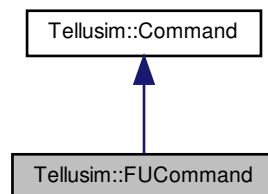
#### 5.135.1 Detailed Description

[FUBuffer](#)

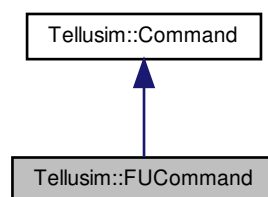
## 5.136 Tellusim::FUCommand Class Reference

```
#include <platform/TellusimCommand.h>
```

Inheritance diagram for Tellusim::FUCommand:



Collaboration diagram for Tellusim::FUCommand:



## Public Member Functions

- **FUCommand** (const Array< [Command](#) > &commands, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumCommands](#) () const  
*Fusion commands.*
- const [Command](#) **getCommand** (uint32\_t index) const
- [Command](#) **getCommand** (uint32\_t index)

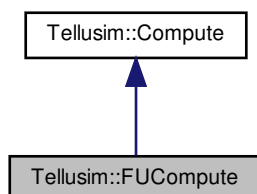
## 5.136.1 Detailed Description

[FUCommand](#)

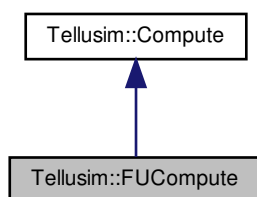
### 5.137 Tellusim::FUCompute Class Reference

```
#include <platform/TellusimCompute.h>
```

Inheritance diagram for Tellusim::FUCompute:



Collaboration diagram for Tellusim::FUCompute:



#### Public Member Functions

- **FUCompute** (const Array< [Compute](#) > &computes, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumComputes](#) () const  
*Fusion computes.*
- const [Compute](#) **getCompute** (uint32\_t index) const
- [Compute](#) **getCompute** (uint32\_t index)

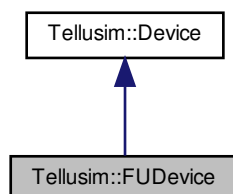
#### 5.137.1 Detailed Description

[FUCompute](#)

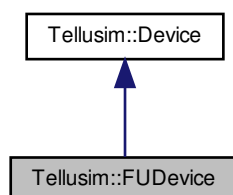
## 5.138 Tellusim::FUDevice Class Reference

```
#include <platform/TellusimDevice.h>
```

Inheritance diagram for Tellusim::FUDevice:



Collaboration diagram for Tellusim::FUDevice:



### Public Member Functions

- **FUDevice** (const Array< [Device](#) > &devices, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumDevices](#) () const  
*Fusion devices.*
- const [Device](#) **getDevice** (uint32\_t index) const
- [Device](#) **getDevice** (uint32\_t index)

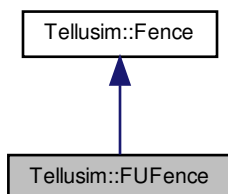
#### 5.138.1 Detailed Description

[FUDevice](#)

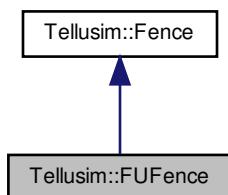
### 5.139 Tellusim::FUFence Class Reference

```
#include <platform/TellusimFence.h>
```

Inheritance diagram for Tellusim::FUFence:



Collaboration diagram for Tellusim::FUFence:



#### Public Member Functions

- **FUFence** (const Array< [Fence](#) > &fences, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumFences](#) () const  
*Fusion fences.*
- const [Fence](#) **getFence** (uint32\_t index) const
- [Fence](#) **getFence** (uint32\_t index)

#### Additional Inherited Members

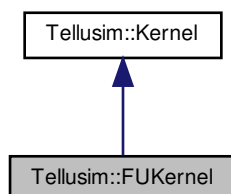
##### 5.139.1 Detailed Description

#### [FUFence](#)

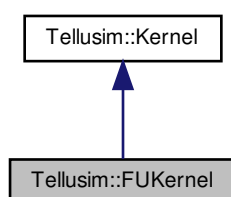
## 5.140 Tellusim::FUKernel Class Reference

```
#include <platform/TellusimKernel.h>
```

Inheritance diagram for Tellusim::FUKernel:



Collaboration diagram for Tellusim::FUKernel:



### Public Member Functions

- **FUKernel** (const Array< [Kernel](#) > &kernels, bool owner=false)
- void **setMask** (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t **getNumKernels** () const  
*Fusion kernels.*
- const [Kernel](#) **getKernel** (uint32\_t index) const
- [Kernel](#) **getKernel** (uint32\_t index)

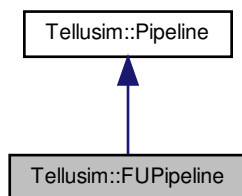
#### 5.140.1 Detailed Description

[FUKernel](#)

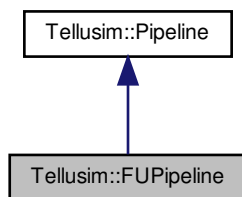
### 5.141 Tellusim::FUPipeline Class Reference

```
#include <platform/TellusimPipeline.h>
```

Inheritance diagram for Tellusim::FUPipeline:



Collaboration diagram for Tellusim::FUPipeline:



#### Public Member Functions

- **FUPipeline** (const Array< [Pipeline](#) > &pipelines, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumPipelines](#) () const  
*Fusion pipelines.*
- const [Pipeline](#) **getPipeline** (uint32\_t index) const
- [Pipeline](#) **getPipeline** (uint32\_t index)

#### Additional Inherited Members

##### 5.141.1 Detailed Description

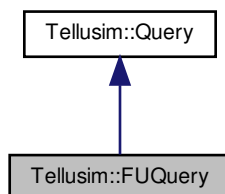
[FUPipeline](#)



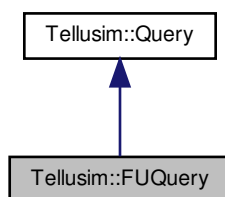
## 5.142 Tellusim::FUQuery Class Reference

```
#include <platform/TellusimQuery.h>
```

Inheritance diagram for Tellusim::FUQuery:



Collaboration diagram for Tellusim::FUQuery:



## Public Member Functions

- **FUQuery** (const Array< [Query](#) > &queries, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumQueries](#) () const  
*Fusion queries.*
- const [Query](#) **getQuery** (uint32\_t index) const
- [Query](#) **getQuery** (uint32\_t index)

## Additional Inherited Members

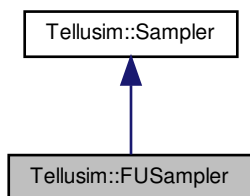
## 5.142.1 Detailed Description

[FUQuery](#)

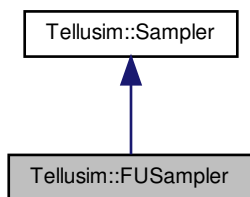
### 5.143 Tellusim::FUSampler Class Reference

```
#include <platform/TellusimSampler.h>
```

Inheritance diagram for Tellusim::FUSampler:



Collaboration diagram for Tellusim::FUSampler:



#### Public Member Functions

- **FUSampler** (const Array< [Sampler](#) > &samplers, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
    *device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumSamplers](#) () const  
    *Fusion samplers.*
- const [Sampler](#) **getSampler** (uint32\_t index) const
- [Sampler](#) **getSampler** (uint32\_t index)

#### Additional Inherited Members

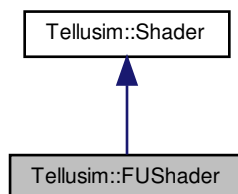
##### 5.143.1 Detailed Description

#### [FUSampler](#)

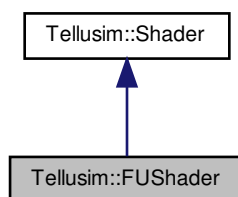
## 5.144 Tellusim::FUShader Class Reference

```
#include <platform/TellusimShader.h>
```

Inheritance diagram for Tellusim::FUShader:



Collaboration diagram for Tellusim::FUShader:



### Public Member Functions

- **FUShader** (const Array< [Shader](#) > &shaders, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumShaders](#) () const  
*Fusion shaders.*
- const [Shader](#) **getShader** (uint32\_t index) const
- [Shader](#) **getShader** (uint32\_t index)

### Additional Inherited Members

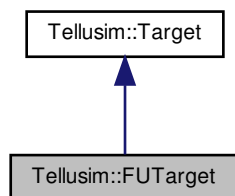
#### 5.144.1 Detailed Description

#### [FUShader](#)

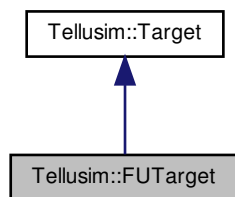
### 5.145 Tellusim::FUTarget Class Reference

```
#include <platform/TellusimTarget.h>
```

Inheritance diagram for Tellusim::FUTarget:



Collaboration diagram for Tellusim::FUTarget:



#### Public Member Functions

- **FUTarget** (const Array< [Target](#) > &targets, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumTargets](#) () const  
*Fusion targets.*
- const [Target](#) **getTarget** (uint32\_t index) const
- [Target](#) **getTarget** (uint32\_t index)

#### Additional Inherited Members

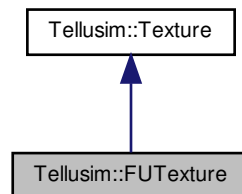
##### 5.145.1 Detailed Description

#### [FUTarget](#)

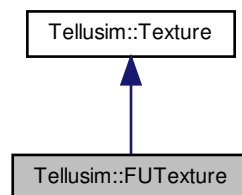
## 5.146 Tellusim::FUTexture Class Reference

```
#include <platform/TellusimTexture.h>
```

Inheritance diagram for Tellusim::FUTexture:



Collaboration diagram for Tellusim::FUTexture:



### Public Member Functions

- **FUTexture** (const Array< [Texture](#) > &textures, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumTextures](#) () const  
*Fusion textures.*
- const [Texture](#) **getTexture** (uint32\_t index) const
- [Texture](#) **getTexture** (uint32\_t index)

### Additional Inherited Members

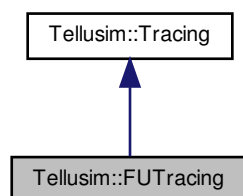
#### 5.146.1 Detailed Description

#### [FUTexture](#)

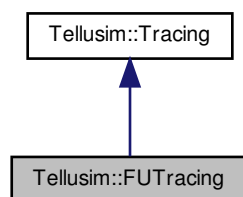
### 5.147 Tellusim::FUTracing Class Reference

```
#include <platform/TellusimTracing.h>
```

Inheritance diagram for Tellusim::FUTracing:



Collaboration diagram for Tellusim::FUTracing:



#### Public Member Functions

- **FUTracing** (const Array< [Tracing](#) > &tracings, bool owner=false)
- void [setMask](#) (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t [getNumTracings](#) () const  
*Fusion tracings.*
- const [Tracing](#) **getTracing** (uint32\_t index) const
- [Tracing](#) **getTracing** (uint32\_t index)

#### Additional Inherited Members

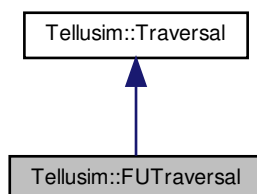
##### 5.147.1 Detailed Description

#### [FUTracing](#)

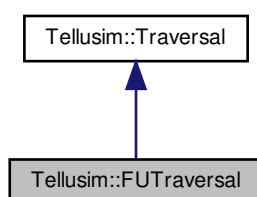
## 5.148 Tellusim::FUTraversal Class Reference

```
#include <platform/TellusimTraversal.h>
```

Inheritance diagram for Tellusim::FUTraversal:



Collaboration diagram for Tellusim::FUTraversal:



## Public Member Functions

- **FUTraversal** (const Array< [Traversal](#) > &traversals, bool owner=false)
- void **setMask** (uint32\_t mask)  
*device mask*
- uint32\_t **getMask** () const
- uint32\_t **getNumTraversals** () const  
*Fusion traversals.*
- const [Traversal](#) **getTraversal** (uint32\_t index) const
- [Traversal](#) **getTraversal** (uint32\_t index)

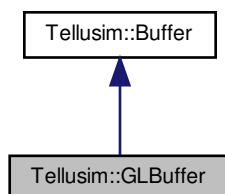
## 5.148.1 Detailed Description

[FUTraversal](#)

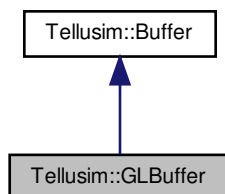
### 5.149 Tellusim::GLBuffer Class Reference

```
#include <platform/TellusimBuffer.h>
```

Inheritance diagram for Tellusim::GLBuffer:



Collaboration diagram for Tellusim::GLBuffer:



#### Public Member Functions

- bool **create** ([Flags](#) flags, uint32\_t target, uint32\_t buffer\_id)  
*create external buffer*
- uint32\_t **getTarget** () const
- uint32\_t **getBufferID** () const

#### Additional Inherited Members

##### 5.149.1 Detailed Description

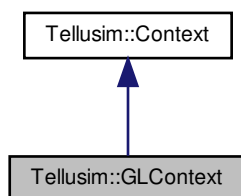
[GLBuffer](#)



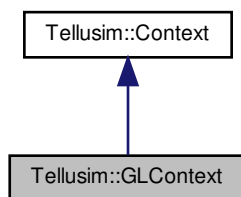
## 5.150 Tellusim::GLContext Class Reference

```
#include <platform/TellusimContext.h>
```

Inheritance diagram for Tellusim::GLContext:



Collaboration diagram for Tellusim::GLContext:



### Public Member Functions

- bool **create** (void \*context)  
*create context*
- void \* **getGLDisplay** () const  
*current context*
- void \* **getGLVisual** () const
- void \* **getGLConfig** () const
- void \* **getGLSurface** () const
- void \* **getGLContext** () const

### Static Public Member Functions

- static void \* **getProcAddress** (const char \*name)  
*OpenGL functions.*
- static bool **error** (uint32\_t result)  
*check OpenGL errors*
- static bool **check** ()

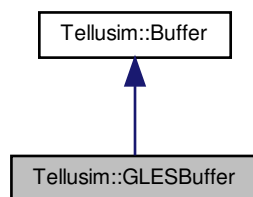
### 5.150.1 Detailed Description

[GLContext](#)

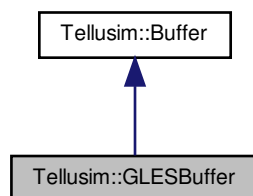
## 5.151 Tellusim::GLESBuffer Class Reference

```
#include <platform/TellusimBuffer.h>
```

Inheritance diagram for Tellusim::GLESBuffer:



Collaboration diagram for Tellusim::GLESBuffer:



### Public Member Functions

- bool [create](#) ([Flags](#) flags, uint32\_t target, uint32\_t buffer\_id)  
*create external buffer*
- uint32\_t **getTarget** () const
- uint32\_t **getBufferID** () const

### Additional Inherited Members

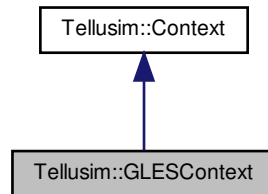
### 5.151.1 Detailed Description

[GLESBuffer](#)

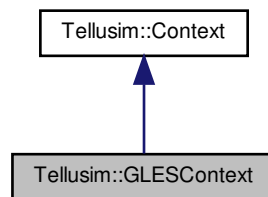
## 5.152 Tellusim::GLESContext Class Reference

```
#include <platform/TellusimContext.h>
```

Inheritance diagram for Tellusim::GLESContext:



Collaboration diagram for Tellusim::GLESContext:



### Public Member Functions

- bool **create** (void \*context)  
*create context*
- void \* **getGLESDisplay** () const  
*current context*
- void \* **getGLESConfig** () const
- void \* **getGLESContext** () const

### Static Public Member Functions

- static void \* **getProcAddress** (const char \*name)  
*OpenGL ES functions.*
- static bool **error** (uint32\_t result)  
*check OpenGL ES errors*
- static bool **check** ()

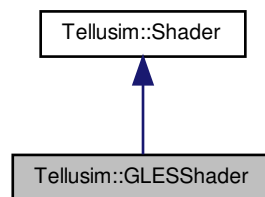
### 5.152.1 Detailed Description

[GLESText](#)

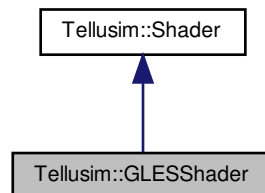
## 5.153 Tellusim::GLESThader Class Reference

```
#include <platform/TellusimShader.h>
```

Inheritance diagram for Tellusim::GLESThader:



Collaboration diagram for Tellusim::GLESThader:



### Public Member Functions

- bool **attachShader** (uint32\_t program\_id)
- uint32\_t **getShaderType** () const
- uint32\_t **getShaderID** () const

### Additional Inherited Members

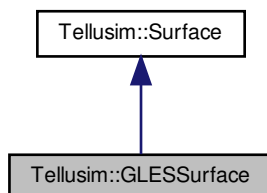
### 5.153.1 Detailed Description

[GLESThader](#)

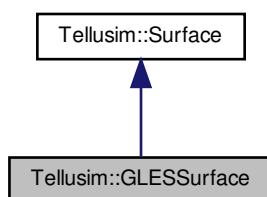
## 5.154 Tellusim::GLESSurface Class Reference

```
#include <platform/TellusimSurface.h>
```

Inheritance diagram for Tellusim::GLESSurface:



Collaboration diagram for Tellusim::GLESSurface:



## Public Member Functions

- **GLESSurface** ([GLESTexture](#) &context)
- void \* [getContext](#) () const  
*current context*
- void [setColorTextureID](#) (uint32\_t texture\_id)  
*texture identifiers*
- void [setDepthTextureID](#) (uint32\_t texture\_id)
- uint32\_t [getColorTextureID](#) () const
- uint32\_t [getDepthTextureID](#) () const
- void [setFramebufferID](#) (uint32\_t framebuffer\_id)  
*framebuffer identifier*
- uint32\_t [getFramebufferID](#) () const
- uint32\_t [getColorInternalFormat](#) () const  
*surface formats*
- uint32\_t [getDepthInternalFormat](#) () const

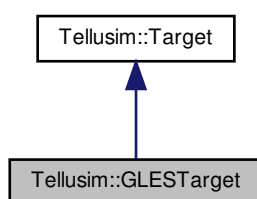
### 5.154.1 Detailed Description

[GLESSurface](#)

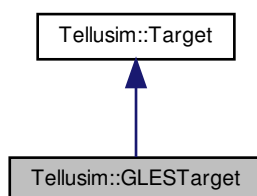
## 5.155 Tellusim::GLESTarget Class Reference

```
#include <platform/TellusimTarget.h>
```

Inheritance diagram for Tellusim::GLESTarget:



Collaboration diagram for Tellusim::GLESTarget:



### Public Member Functions

- `uint32_t getFramebufferID () const`

### Additional Inherited Members

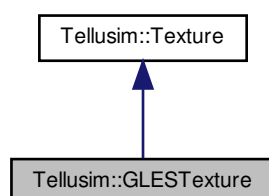
### 5.155.1 Detailed Description

[GLESTarget](#)

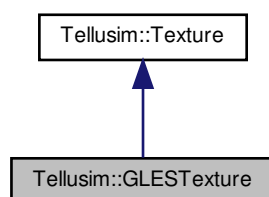
## 5.156 Tellusim::GLESTexture Class Reference

```
#include <platform/TellusimTexture.h>
```

Inheritance diagram for Tellusim::GLESTexture:



Collaboration diagram for Tellusim::GLESTexture:



### Public Member Functions

- bool [create](#) (uint32\_t target, uint32\_t texture\_id, [Flags](#) flags=DefaultFlags, Format format=FormatUnknown)  
*create external texture*
- uint32\_t **getTarget** () const
- uint32\_t **getInternalFormat** () const
- uint32\_t **getTextureID** () const

### Additional Inherited Members

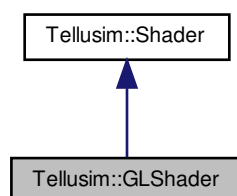
#### 5.156.1 Detailed Description

[GLESTexture](#)

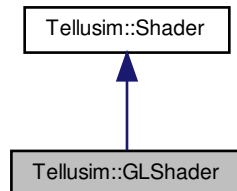
### 5.157 Tellusim::GLShader Class Reference

```
#include <platform/TellusimShader.h>
```

Inheritance diagram for Tellusim::GLShader:



Collaboration diagram for Tellusim::GLShader:



#### Public Member Functions

- bool **attachShader** (uint32\_t program\_id)
- uint32\_t **getShaderType** () const
- uint32\_t **getShaderID** () const

#### Additional Inherited Members

#### 5.157.1 Detailed Description

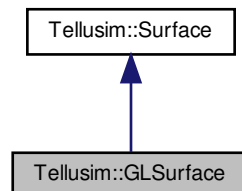
[GLShader](#)



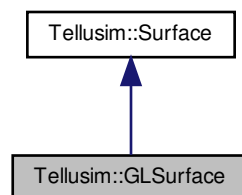
## 5.158 Tellusim::GLSurface Class Reference

```
#include <platform/TellusimSurface.h>
```

Inheritance diagram for Tellusim::GLSurface:



Collaboration diagram for Tellusim::GLSurface:



## Public Member Functions

- **GLSurface** ([GLContext](#) &context)
- void \* [getContext](#) () const  
*current context*
- void [setColorTextureID](#) (uint32\_t texture\_id)  
*texture identifiers*
- void [setDepthTextureID](#) (uint32\_t texture\_id)
- uint32\_t [getColorTextureID](#) () const
- uint32\_t [getDepthTextureID](#) () const
- void [setFramebufferID](#) (uint32\_t framebuffer\_id)  
*framebuffer identifier*
- uint32\_t [getFramebufferID](#) () const
- uint32\_t [getColorInternalFormat](#) () const  
*surface formats*
- uint32\_t [getDepthInternalFormat](#) () const

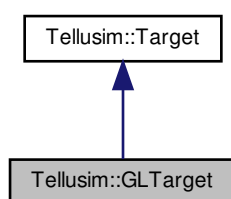
### 5.158.1 Detailed Description

[GLSurface](#)

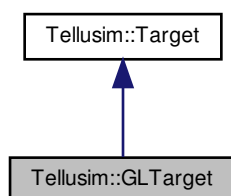
## 5.159 Tellusim::GLTarget Class Reference

```
#include <platform/TellusimTarget.h>
```

Inheritance diagram for Tellusim::GLTarget:



Collaboration diagram for Tellusim::GLTarget:



### Public Member Functions

- `uint32_t getFramebufferID () const`

### Additional Inherited Members

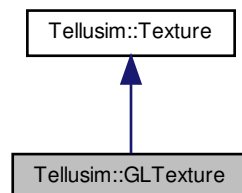
### 5.159.1 Detailed Description

[GLTarget](#)

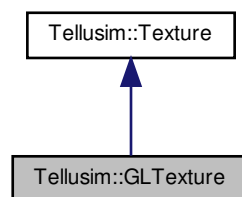
## 5.160 Tellusim::GLTexture Class Reference

```
#include <platform/TellusimTexture.h>
```

Inheritance diagram for Tellusim::GLTexture:



Collaboration diagram for Tellusim::GLTexture:



### Public Member Functions

- bool [create](#) (uint32\_t target, uint32\_t texture\_id, [Flags](#) flags=DefaultFlags, Format format=FormatUnknown)  
*create external texture*
- uint32\_t **getTarget** () const
- uint32\_t **getInternalFormat** () const
- uint32\_t **getTextureID** () const

### Additional Inherited Members

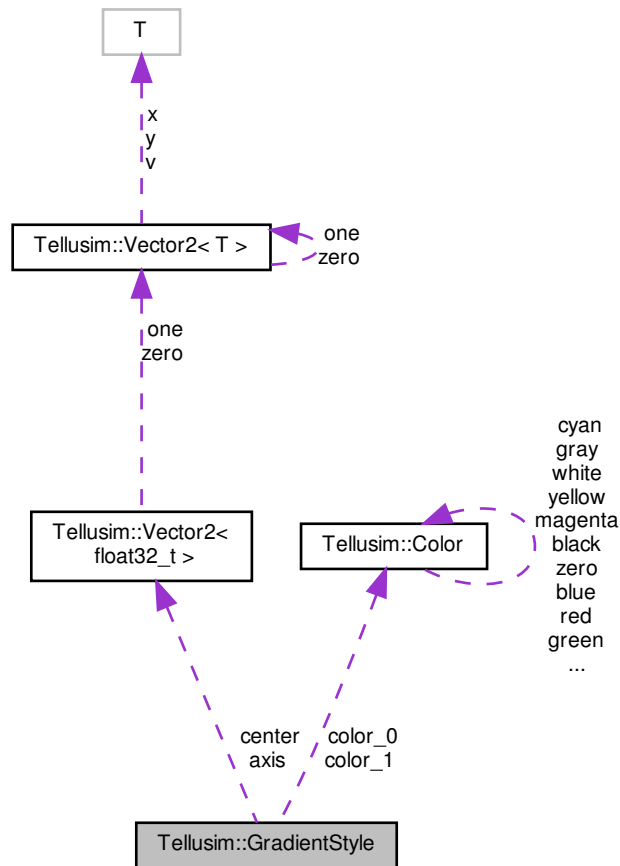
#### 5.160.1 Detailed Description

[GLTexture](#)

### 5.161 Tellusim::GradientStyle Struct Reference

```
#include <interface/TellusimTypes.h>
```

Collaboration diagram for Tellusim::GradientStyle:



#### Public Member Functions

- **GradientStyle** (const [Color](#) &c0, const [Color](#) &c1)
- **GradientStyle** (float32\_t radius, const [Vector2f](#) &center)
- **GradientStyle** (float32\_t length, const [Vector2f](#) &center, const [Vector2f](#) &axis)
- **GradientStyle** (float32\_t radius, const [Vector2f](#) &center, const [Color](#) &c0, const [Color](#) &c1)
- **GradientStyle** (float32\_t length, const [Vector2f](#) &center, const [Vector2f](#) &axis, const [Color](#) &c0, const [Color](#) &c1)
- bool **isValid** () const  
*check style*
- **operator bool** () const

## Public Attributes

- float32\_t **radius** = 0.0f
- float32\_t **length** = 0.0f
- [Vector2f](#) **center** = [Vector2f::zero](#)
- [Vector2f](#) **axis** = [Vector2f::zero](#)
- [Color](#) **color\_0** = [Color::white](#)
- [Color](#) **color\_1** = [Color::black](#)

## 5.161.1 Detailed Description

[GradientStyle](#)

## 5.162 Tellusim::HeapPool&lt; Threshold &gt; Class Template Reference

```
#include <core/TellusimPool.h>
```

## Public Member Functions

- [HeapPool](#) ()  
*constructor*
- **HeapPool** (uint32\_t size)
- void [init](#) (uint32\_t size)  
*initialize pool*
- void [shutdown](#) ()  
*shutdown pool*
- bool [isInitialized](#) ()  
*pool status*
- void [append](#) (uint32\_t size)  
*append memory*
- uint32\_t [allocate](#) (uint32\_t alignment, uint32\_t size)  
*allocate memory*
- void [free](#) (uint32\_t offset)  
*free memory*
- uint32\_t [getMemory](#) () const  
*memory usage in bytes*
- uint32\_t [getAllocations](#) () const  
*number of allocations*
- uint32\_t [getBlocks](#) () const  
*heap fragmentation*

## 5.162.1 Detailed Description

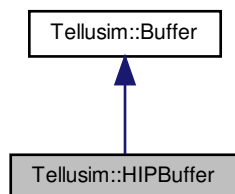
```
template<uint32_t Threshold = 8>
class Tellusim::HeapPool< Threshold >
```

Heap pool memory allocator

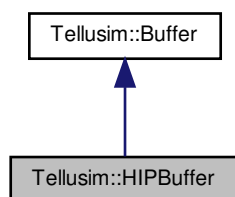
### 5.163 Tellusim::HIPBuffer Class Reference

```
#include <platform/TellusimBuffer.h>
```

Inheritance diagram for Tellusim::HIPBuffer:



Collaboration diagram for Tellusim::HIPBuffer:



#### Public Member Functions

- void \* **getBufferPtr** () const
- uint8\_t \* **getBufferData** () const
- void \* **getBufferEvent** () const
- uint32\_t **getArrayFormat** () const
- uint32\_t **getArrayChannels** () const
- void \* **getSharedMemory** () const

#### Additional Inherited Members

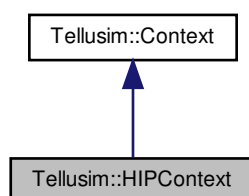
##### 5.163.1 Detailed Description

[HIPBuffer](#)

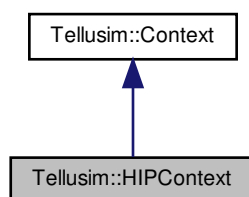
## 5.164 Tellusim::HIPContext Class Reference

```
#include <platform/TellusimContext.h>
```

Inheritance diagram for Tellusim::HIPContext:



Collaboration diagram for Tellusim::HIPContext:



#### Public Member Functions

- `int32_t` [getDevice](#) () const  
*current device*
- `void *` **getHIPContext** () const
- `void *` **getStream** () const

#### Static Public Member Functions

- `static void *` [getProcAddress](#) (const char \*name)  
*Hip functions.*
- `static bool` [error](#) (uint32\_t result)  
*check Hip errors*

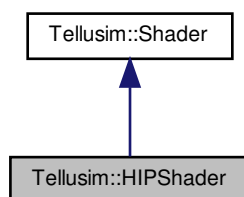
### 5.164.1 Detailed Description

[HIPContext](#)

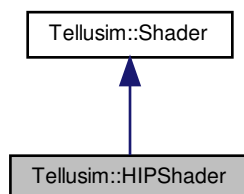
### 5.165 Tellusim::HIPShader Class Reference

```
#include <platform/TellusimShader.h>
```

Inheritance diagram for Tellusim::HIPShader:



Collaboration diagram for Tellusim::HIPShader:



#### Public Member Functions

- void \* **getModule** () const
- void \* **getFunction** () const

#### Additional Inherited Members

### 5.165.1 Detailed Description

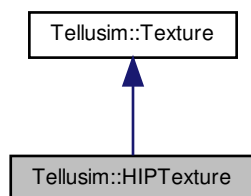
[HIPShader](#)



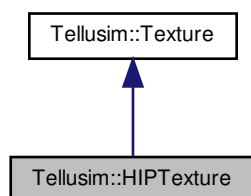
## 5.166 Tellusim::HIPTexture Class Reference

```
#include <platform/TellusimTexture.h>
```

Inheritance diagram for Tellusim::HIPTexture:



Collaboration diagram for Tellusim::HIPTexture:



### Public Member Functions

- void \* **getTextureArray** () const
- void \* **getTextureLevel** (uint32\_t index) const
- const void \* **getChannelFormat** () const

### Additional Inherited Members

#### 5.166.1 Detailed Description

[HIPTexture](#)

## 5.167 Tellusim::Image Class Reference

```
#include <format/TellusimImage.h>
```

## Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagMipmaps** = (1 << 0),  
**FlagNoClear** = (1 << 1),  
**FlagNoAllocate** = (1 << 2),  
**FlagFast** = (1 << 3),  
**FlagBest** = (1 << 4),  
**FlagPerceptual** = (1 << 5),  
**FlagPanorama** = (1 << 6),  
**FlagNormalize** = (1 << 7),  
**FlagGamma** = (1 << 8),  
**FlagSRGB** = (1 << 9),  
**NumFlags** = 10 }

[Image](#) flags.

- enum [Filter](#) {  
**FilterUnknown** = 0,  
**FilterPoint**,  
**FilterLinear**,  
**FilterCubic**,  
**FilterSinc**,  
**FilterBox**,  
**FilterMax**,  
**FilterMin**,  
**FilterMip**,  
**FilterCR**,  
**NumFilters** }

[Image](#) filters.

## Public Member Functions

- **Image** (const char \*name, [Flags](#) flags=FlagNone, uint32\_t offset=0)
- **Image** ([Stream](#) &stream, [Flags](#) flags=FlagNone, uint32\_t offset=0)
- **Image** (Type type, Format format, const [Size](#) &size, [Flags](#) flags=FlagNone)
- **Image** (Type type, Format format, const [Size](#) &size, uint32\_t layers, [Flags](#) flags=FlagNone)
- void [clear](#) ()  
*clear image*
- bool [isLoading](#) () const  
*check image*
- bool [isAllocated](#) () const
- Type [getType](#) () const  
*image type*
- const char \* [getTypeName](#) () const
- bool [is2DType](#) () const
- bool [is3DType](#) () const
- bool [isCubeType](#) () const
- Format [getFormat](#) () const  
*image format*
- const char \* [getFormatName](#) () const
- bool [isColorFormat](#) () const
- bool [isDepthFormat](#) () const
- bool [isPixelFormat](#) () const
- bool [isPlainFormat](#) () const

- bool **isMixedFormat** () const
  - bool **isBlockFormat** () const
  - bool **isStencilFormat** () const
  - bool **isNormFormat** () const
  - bool **isSRGBFormat** () const
  - bool **isFloatFormat** () const
  - bool **isSignedFormat** () const
  - bool **isUnsignedFormat** () const
  - bool **isIntegerFormat** () const
  - bool **isi8Format** () const
  - bool **isu8Format** () const
  - bool **is8BitFormat** () const
  - bool **isi16Format** () const
  - bool **isu16Format** () const
  - bool **isf16Format** () const
  - bool **is16BitFormat** () const
  - bool **isi32Format** () const
  - bool **isu32Format** () const
  - bool **isf32Format** () const
  - bool **is32BitFormat** () const
  - bool **isi64Format** () const
  - bool **isu64Format** () const
  - bool **isf64Format** () const
  - bool **is64BitFormat** () const
  - bool **isBC15Format** () const
  - bool **isBC67Format** () const
  - bool **isETC2Format** () const
  - bool **isASTCFormat** () const
  - uint32\_t **getComponents** () const
  - uint32\_t **getPixelSize** () const
  - uint32\_t **getBlockSize** () const
  - uint32\_t **getBlockWidth** () const
  - uint32\_t **getBlockHeight** () const
  - uint32\_t **getWidth** () const
- image dimension*
- uint32\_t **getHeight** () const
  - uint32\_t **getDepth** () const
  - uint32\_t **getFaces** () const
  - uint32\_t **getLayers** () const
  - uint32\_t **getMipmaps** () const
  - uint32\_t **findMipmap** (const [Size](#) &size) const
  - uint32\_t **getWidth** (uint32\_t mipmap) const
  - uint32\_t **getHeight** (uint32\_t mipmap) const
  - uint32\_t **getDepth** (uint32\_t mipmap) const
  - bool **hasLayers** () const
  - bool **hasMipmaps** () const
  - [Size](#) **getSize** () const
  - [Region](#) **getRegion** () const
  - [Slice](#) **getSlice** () const
  - [Size](#) **getSize** (uint32\_t mipmap) const
  - [Region](#) **getRegion** (uint32\_t mipmap) const
  - [Slice](#) **getSlice** (uint32\_t mipmap) const
  - void **setMetaInfo** (const [String](#) &str)

*image meta info*

- [String](#) **getMetalInfo** () const
- [String](#) **getDescription** () const
  - image description*
- [size\\_t](#) **getOffset** (const [Slice](#) &slice, [uint32\\_t](#) alignment=1) const
  - image layout*
- [size\\_t](#) **getStride** ([uint32\\_t](#) mipmap=0, [uint32\\_t](#) alignment=1) const
- [size\\_t](#) **getMipmapSize** ([uint32\\_t](#) mipmap, [uint32\\_t](#) alignment=1) const
- [size\\_t](#) **getLayerSize** ([uint32\\_t](#) alignment=1) const
- [size\\_t](#) **getDataSize** ([uint32\\_t](#) alignment=1) const
- [bool](#) **create** (Type type, Format format, const [Size](#) &size, [Flags](#) flags=FlagNone)
  - create image*
- [bool](#) **create** (Type type, Format format, const [Size](#) &size, [uint32\\_t](#) layers, [Flags](#) flags=FlagNone)
- [bool](#) **create2D** (Format format, [uint32\\_t](#) size, [Flags](#) flags=FlagNone)
- [bool](#) **create3D** (Format format, [uint32\\_t](#) size, [Flags](#) flags=FlagNone)
- [bool](#) **createCube** (Format format, [uint32\\_t](#) size, [Flags](#) flags=FlagNone)
- [bool](#) **create2D** (Format format, [uint32\\_t](#) width, [uint32\\_t](#) height, [Flags](#) flags=FlagNone)
- [bool](#) **create3D** (Format format, [uint32\\_t](#) width, [uint32\\_t](#) height, [uint32\\_t](#) depth, [Flags](#) flags=FlagNone)
- [bool](#) **create2D** (Format format, [uint32\\_t](#) width, [uint32\\_t](#) height, [uint32\\_t](#) layers, [Flags](#) flags=FlagNone)
- [bool](#) **createCube** (Format format, [uint32\\_t](#) size, [uint32\\_t](#) layers, [Flags](#) flags=FlagNone)
- [bool](#) **info** (const char \*name, [Flags](#) flags=FlagNone, [uint32\\_t](#) offset=0, [Async](#) \*async=NULLPTR)
  - info image*
- [bool](#) **info** (const [String](#) &name, [Flags](#) flags=FlagNone, [uint32\\_t](#) offset=0, [Async](#) \*async=NULLPTR)
- [bool](#) **info** ([Stream](#) &stream, [Flags](#) flags=FlagNone, [uint32\\_t](#) offset=0, [Async](#) \*async=NULLPTR)
- [bool](#) **info** (const char \*name, [Async](#) \*async)
- [bool](#) **info** (const [String](#) &name, [Async](#) \*async)
- [bool](#) **info** ([Stream](#) &stream, [Async](#) \*async)
- [bool](#) **load** (const char \*name, [Flags](#) flags=FlagNone, [uint32\\_t](#) offset=0, [Async](#) \*async=NULLPTR)
  - load image*
- [bool](#) **load** (const [String](#) &name, [Flags](#) flags=FlagNone, [uint32\\_t](#) offset=0, [Async](#) \*async=NULLPTR)
- [bool](#) **load** ([Stream](#) &stream, [Flags](#) flags=FlagNone, [uint32\\_t](#) offset=0, [Async](#) \*async=NULLPTR)
- [bool](#) **load** (const char \*name, [Async](#) \*async)
- [bool](#) **load** (const [String](#) &name, [Async](#) \*async)
- [bool](#) **load** ([Stream](#) &stream, [Async](#) \*async)
- [bool](#) **save** (const char \*name, [Flags](#) flags=FlagNone, [uint32\\_t](#) quality=95) const
  - save image*
- [bool](#) **save** (const [String](#) &name, [Flags](#) flags=FlagNone, [uint32\\_t](#) quality=95) const
- [bool](#) **save** ([Stream](#) &stream, [Flags](#) flags=FlagNone, [uint32\\_t](#) quality=95) const
- [bool](#) **swap** ([uint32\\_t](#) component\_0, [uint32\\_t](#) component\_1)
  - image components*
- [bool](#) **copy** (const [Image](#) &src, [uint32\\_t](#) dest\_component, [uint32\\_t](#) src\_component)
- [bool](#) **flipX** (const [Region](#) &region, const [Slice](#) &slice)
  - flip horizontally*
- [bool](#) **flipX** (const [Region](#) &region)
- [bool](#) **flipX** ()
- [bool](#) **flipY** (const [Region](#) &region, const [Slice](#) &slice)
  - flip vertically*
- [bool](#) **flipY** (const [Region](#) &region)
- [bool](#) **flipY** ()
- [bool](#) **copy** (const [Image](#) &src, const [Origin](#) &dest\_origin, const [Region](#) &src\_region, const [Slice](#) &dest\_slice, const [Slice](#) &src\_slice)
  - copy image*
- [bool](#) **copy** (const [Image](#) &src, const [Origin](#) &dest\_origin, const [Region](#) &src\_region)
- [bool](#) **copy** (const [Image](#) &src, const [Origin](#) &dest\_origin, const [Slice](#) &dest\_slice)

- bool **copy** (const [Image](#) &src, const [Slice](#) &dest\_slice, const [Slice](#) &src\_slice)
- bool **copy** (const [Image](#) &src, const [Origin](#) &dest\_origin)
- bool **copy** (const [Image](#) &src, const [Slice](#) &dest\_slice)
- [Image toType](#) (Type type, [Flags](#) flags, [Async](#) \*async=nullptr) const  
*convert image to type*
- [Image toType](#) (Type type, [Async](#) \*async=nullptr) const
- [Image toFormat](#) (Format format, [Flags](#) flags, [Async](#) \*async=nullptr) const  
*convert image to format*
- [Image toFormat](#) (Format format, [Async](#) \*async=nullptr) const
- [Image getSlice](#) (const [Slice](#) &slice) const  
*get image slice*
- [Image getComponent](#) (uint32\_t component) const  
*get image component*
- [Image getRegion](#) (const [Region](#) &region, const [Slice](#) &slice) const  
*get image region*
- [Image getRegion](#) (const [Region](#) &region) const
- [Image getRotated](#) (int32\_t angle, const [Slice](#) &slice) const  
*get rotated image*
- [Image getRotated](#) (int32\_t angle) const
- [Image getResized](#) (const [Size](#) &size, [Filter](#) min, [Filter](#) mag, [Flags](#) flags, [Async](#) \*async=nullptr) const  
*get resized image*
- [Image getResized](#) (const [Size](#) &size, [Filter](#) min, [Filter](#) mag=FilterCubic, [Async](#) \*async=nullptr) const
- [Image getResized](#) (const [Size](#) &size, [Async](#) \*async=nullptr) const
- [Image getMipmapped](#) ([Filter](#) filter, [Flags](#) flags, [Async](#) \*async=nullptr) const  
*get mipmapped image*
- [Image getMipmapped](#) ([Filter](#) filter, [Async](#) \*async=nullptr) const
- [Image getMipmapped](#) ([Async](#) \*async=nullptr) const
- int32\_t **compare** (const [Image](#) &image) const  
*compare images*
- const uint8\_t \* **getData** (const [Slice](#) &slice=[Slice](#)()) const  
*image data*
- uint8\_t \* **getData** (const [Slice](#) &slice=[Slice](#)())
- const uint8\_t \* **getData** (const [Origin](#) &origin, const [Slice](#) &slice=[Slice](#)()) const
- uint8\_t \* **getData** (const [Origin](#) &origin, const [Slice](#) &slice=[Slice](#)())
- bool **setData** (const void \*src, const [Slice](#) &slice=[Slice](#)(), uint32\_t alignment=1, size\_t stride=0)
- bool **getData** (void \*dest, const [Slice](#) &slice=[Slice](#)(), uint32\_t alignment=1, size\_t stride=0) const
- size\_t **getMemory** () const  
*memory usage*

#### Static Public Member Functions

- static const char \* **getTypeName** (Type type)

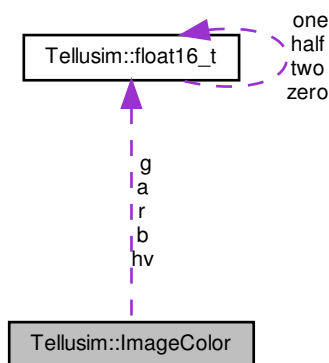
#### 5.167.1 Detailed Description

[Image](#) class

## 5.168 Tellusim::ImageColor Struct Reference

```
#include <format/TellusimImage.h>
```

Collaboration diagram for Tellusim::ImageColor:



### Public Types

- enum { **Size** = 4 }

### Public Member Functions

- **ImageColor** (const [ImageColor](#) &c)
- **ImageColor** (int32\_t i)
- **ImageColor** (uint32\_t u)
- **ImageColor** ([float16\\_t](#) h)
- **ImageColor** (float32\_t f)
- **ImageColor** (int32\_t l, int32\_t a)
- **ImageColor** (uint32\_t l, uint32\_t a)
- **ImageColor** ([float16\\_t](#) l, [float16\\_t](#) a)
- **ImageColor** (float32\_t l, float32\_t a)
- **ImageColor** (int32\_t r, int32\_t g, int32\_t b, int32\_t a)
- **ImageColor** (uint32\_t r, uint32\_t g, uint32\_t b, uint32\_t a)
- **ImageColor** ([float16\\_t](#) r, [float16\\_t](#) g, [float16\\_t](#) b, [float16\\_t](#) a)
- **ImageColor** (float32\_t r, float32\_t g, float32\_t b, float32\_t a)
- **ImageColor** (const [Color](#) &color, Format format)
- void **set** (const [Color](#) &color, Format format)
- *color value*
- [Color](#) **get** (Format format) const
- [ImageColor](#) & **operator=** (const [ImageColor](#) &c)
- *assignment operator*

## Public Attributes

- ```

union {
    struct {
        int32_t r
        int32_t g
        int32_t b
        int32_t a
    } i
    struct {
        uint32_t r
        uint32_t g
        uint32_t b
        uint32_t a
    } u
    struct {
        float16_t r
        float16_t g
        float16_t b
        float16_t a
    } h
    struct {
        float32_t r
        float32_t g
        float32_t b
        float32_t a
    } f
    int32_t iv [Size]
    uint32_t uv [Size]
    float16_t hv [Size]
    float32_t fv [Size]
};

```

## 5.168.1 Detailed Description

[ImageColor](#) struct

## 5.169 Tellusim::ImageSampler Class Reference

```
#include <format/TellusimImage.h>
```

## Public Types

- using **Filter** = [Image::Filter](#)

## Public Member Functions

- **ImageSampler** ([Image](#) &image, const [Slice](#) &slice=[Slice](#)())
- **ImageSampler** (const [Image](#) &image, const [Slice](#) &slice=[Slice](#)())
- void **clear** ()
  - clear sampler*
- bool **isCreated** () const
  - check sampler*
- Type **getType** () const
  - sampler type*
- bool **is2DType** () const
- bool **is3DType** () const
- bool **isCubeType** () const
- Format **getFormat** () const
  - sampler format*
- const char \* **getFormatName** () const
- uint32\_t **getWidth** () const
  - sampler dimension*
- uint32\_t **getHeight** () const
- uint32\_t **getDepth** () const
- uint32\_t **getFaces** () const
- size\_t **getTexels** () const
- [Size](#) **getSize** () const
- [Region](#) **getRegion** () const
- size\_t **getStride** () const
  - sampler layout*
- size\_t **getLayerSize** () const
- uint32\_t **getPixelSize** () const
- uint32\_t **getComponents** () const
- const uint8\_t \* **getData** () const
  - sampler data*
- uint8\_t \* **getData** ()
- bool **create** ([Image](#) &image, const [Slice](#) &slice=[Slice](#)())
  - create sampler*
- bool **create** (const [Image](#) &image, const [Slice](#) &slice=[Slice](#)())
- bool **create** (Type type, Format format, const [Size](#) &size, size\_t stride, void \*data)
- bool **create** (Type type, Format format, const [Size](#) &size, size\_t stride, const void \*data)
- bool **create** (Type type, Format format, const [Size](#) &size, size\_t stride, size\_t layer\_size, void \*data)
- bool **create** (Type type, Format format, const [Size](#) &size, size\_t stride, size\_t layer\_size, const void \*data)
- bool **clear** (const [Color](#) &color)
  - clear image*
- bool **clear** (const [ImageColor](#) &color)
- bool **mad** (const [Color](#) &m, const [Color](#) &a)
  - multiply accumulate image*
- void **set2D** (uint32\_t x, uint32\_t y, const [ImageColor](#) &color)
  - 2D image colors*
- [ImageColor](#) **get2D** (uint32\_t x, uint32\_t y, bool repeat=false) const
- [ImageColor](#) **get2D** (float64\_t x, float64\_t y, bool repeat=false, [Filter](#) filter=[Image::FilterLinear](#)) const
- void **set3D** (uint32\_t x, uint32\_t y, uint32\_t z, const [ImageColor](#) &color)
  - 3D image colors*
- [ImageColor](#) **get3D** (uint32\_t x, uint32\_t y, uint32\_t z, bool repeat=false) const
- [ImageColor](#) **get3D** (float32\_t x, float32\_t y, float32\_t z, bool repeat=false, [Filter](#) filter=[Image::FilterLinear](#)) const



- void **setCube** (float32\_t x, float32\_t y, float32\_t z, const [ImageColor](#) &color)  
*Cube image colors.*
- [ImageColor](#) **getCube** (float32\_t x, float32\_t y, float32\_t z, [Filter](#) filter=Image::FilterLinear) const
- uint32\_t **getCubeFace** (float32\_t x, float32\_t y, float32\_t z, float32\_t &tx, float32\_t &ty) const
- void **setTexel** (size\_t t, const [ImageColor](#) &color)  
*Texel image colors.*
- [ImageColor](#) **getTexel** (size\_t t) const

#### 5.169.1 Detailed Description

[ImageSampler](#) class

## 5.170 Tellusim::ImageStream Class Reference

```
#include <format/TellusimImage.h>
```

### Public Member Functions

- virtual bool **info** ([Stream](#) &stream, [Image](#) &image, [Image::Flags](#) flags, uint32\_t offset, [Async](#) \*async)
- virtual bool **load** ([Stream](#) &stream, [Image](#) &image, [Image::Flags](#) flags, uint32\_t offset, [Async](#) \*async)
- virtual bool **save** ([Stream](#) &stream, const [Image](#) &image, [Image::Flags](#) flags, uint32\_t quality)

### Static Public Member Functions

- static bool **check** (const [String](#) &name, uint32\_t magic=0)  
*image stream formats*
- static [String](#) **getLoadFormats** ()  
*list of supported formats*
- static [String](#) **getSaveFormats** ()

### Protected Types

- enum **Flags** {  
    **FlagNone** = 0,  
    **FlagLoad** = (1 << 0),  
    **FlagSave** = (1 << 1),  
    **FlagLoadSave** = (FlagLoad | FlagSave) }

### Protected Member Functions

- **ImageStream** ([Flags](#) flags, const char \*name, uint32\_t magic=0)
- **ImageStream** ([Flags](#) flags, const [InitializerList](#)< const char \*> &names, uint32\_t magic=0)
- **ImageStream** ([Flags](#) flags, const [InitializerList](#)< const char \*> &names, const [InitializerList](#)< uint32\_t > &magics)

## 5.170.1 Detailed Description

[ImageStream](#) class

## 5.171 Tellusim::Info Class Reference

```
#include <core/TellusimSystem.h>
```

## Public Member Functions

- `size_t getSystemMemory () const`  
*System info.*
- `uint64_t getSystemUptime () const`
- `String getSystemName () const`
- `String getSystemVersion () const`
- `String getKernelVersion () const`
- `uint32_t getCPUCount () const`  
*CPU info.*
- `String getCPUName (uint32_t index) const`
- `String getCPUVendor (uint32_t index) const`
- `uint32_t getCPUCores (uint32_t index) const`
- `uint32_t getCPUThreads (uint32_t index) const`
- `uint64_t getCPUFrequency (uint32_t index) const`
- `uint32_t getCPUTemperature (uint32_t index) const`
- `uint32_t getCPUUtilization (uint32_t index) const`
- `uint32_t getCPUFanSpeed (uint32_t index) const`
- `uint32_t getCPUPower (uint32_t index) const`
- `uint32_t getGPUCount () const`  
*GPU info.*
- `String getGPUName (uint32_t index) const`
- `String getGPUVendor (uint32_t index) const`
- `String getGPUSerial (uint32_t index) const`
- `String getGPUDevice (uint32_t index) const`
- `String getGPUVersion (uint32_t index) const`
- `size_t getGPUMemory (uint32_t index) const`
- `uint32_t getGPUScreens (uint32_t index) const`
- `uint64_t getGPUFrequency (uint32_t index) const`
- `uint32_t getGPUTemperature (uint32_t index) const`
- `uint32_t getGPUUtilization (uint32_t index) const`
- `uint32_t getGPUFanSpeed (uint32_t index) const`
- `uint32_t getGPUPower (uint32_t index) const`
- `bool isGPUThrottling (uint32_t index) const`

## 5.171.1 Detailed Description

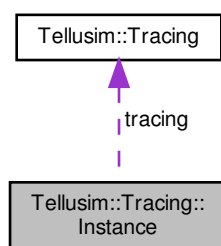
[System Info](#)

## 5.172 Tellusim::Tracing::Instance Struct Reference

tracing instance

```
#include <platform/TellusimTracing.h>
```

Collaboration diagram for Tellusim::Tracing::Instance:



### Public Attributes

- float32\_t **transform** [12]
- uint32\_t **data**
- uint32\_t **mask**
- uint32\_t **flags**
- uint32\_t **offset**
- [Tracing](#) \* **tracing**

### 5.172.1 Detailed Description

tracing instance

## 5.173 Tellusim::int16x8\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

### Public Types

- enum { **Size** = 8 }

## Public Member Functions

- [int16x8\\_t](#) (const [int32x8\\_t](#) &v)
- [int16x8\\_t](#) (const [int16\\_t](#) \*v)
- [int16x8\\_t](#) (const [int32x4\\_t](#) &v0, const [int32x4\\_t](#) &v1)
- [int16x8\\_t](#) ([int16\\_t](#) v)
- [int16x8\\_t](#) ([int16\\_t](#) x0, [int16\\_t](#) y0, [int16\\_t](#) z0, [int16\\_t](#) w0, [int16\\_t](#) x1, [int16\\_t](#) y1, [int16\\_t](#) z1, [int16\\_t](#) w1)
- [int32x4\\_t](#) [asi32x4](#) () const
- cast vector data*
- [uint16x8\\_t](#) [asu16x8](#) () const
- [uint32x4\\_t](#) [asu32x4](#) () const
- [float16x8\\_t](#) [asf16x8](#) () const
- [float32x4\\_t](#) [asf32x4](#) () const
- void [set](#) (const [int16x8\\_t](#) &v)
- update vector data*
- void [set](#) ([int16\\_t](#) X0, [int16\\_t](#) Y0, [int16\\_t](#) Z0, [int16\\_t](#) W0, [int16\\_t](#) X1, [int16\\_t](#) Y1, [int16\\_t](#) Z1, [int16\\_t](#) W1)
- void [set](#) (const [int16\\_t](#) \*1 v)
- void [get](#) ([int16\\_t](#) \*1 v) const
- template<[uint32\\_t](#) Index>  
void [set](#) ([int16\\_t](#) V)
- template<[uint32\\_t](#) Index>  
[int16\\_t](#) [get](#) () const
- [int16x8\\_t](#) & [operator\\*=](#) ([int16\\_t](#) v)
- vector to scalar operators*
- [int16x8\\_t](#) & [operator+=](#) ([int16\\_t](#) v)
- [int16x8\\_t](#) & [operator-=](#) ([int16\\_t](#) v)
- [int16x8\\_t](#) & [operator &=](#) ([int16\\_t](#) v)
- [int16x8\\_t](#) & [operator|=](#) ([int16\\_t](#) v)
- [int16x8\\_t](#) & [operator^=](#) ([int16\\_t](#) v)
- [int16x8\\_t](#) & [operator<=<=](#) ([int16\\_t](#) v)
- [int16x8\\_t](#) & [operator>=>=](#) ([int16\\_t](#) v)
- [int16x8\\_t](#) & [operator\\*=](#) (const [int16x8\\_t](#) &v)
- vector to vector operators*
- [int16x8\\_t](#) & [operator+=](#) (const [int16x8\\_t](#) &v)
- [int16x8\\_t](#) & [operator-=](#) (const [int16x8\\_t](#) &v)
- [int16x8\\_t](#) & [operator &=](#) (const [int16x8\\_t](#) &v)
- [int16x8\\_t](#) & [operator|=](#) (const [int16x8\\_t](#) &v)
- [int16x8\\_t](#) & [operator^=](#) (const [int16x8\\_t](#) &v)
- [int16x8\\_t](#) [xyzw10](#) () const
- swizzle vector*
- [int16x8\\_t](#) [zwxy01](#) () const
- [int16x8\\_t](#) [yxwz01](#) () const
- [int32x4\\_t](#) [xyzw0](#) () const
- swizzle vector*
- [int32x4\\_t](#) [xyzw1](#) () const
- [int16\\_t](#) [sum](#) () const
- sum vector components*

## Public Attributes

- ```

union {
    struct {
        int16_t x0
        int16_t y0
        int16_t z0
        int16_t w0
        int16_t x1
        int16_t y1
        int16_t z1
        int16_t w1
    }
    int16_t v [Size]
};

```

## 5.173.1 Detailed Description

SSE Utils AVX utils NEON utils Vector of eight int16\_t components

## 5.173.2 Constructor &amp; Destructor Documentation

## 5.173.2.1 int16x8\_t()

```

Tellusim::int16x8_t::int16x8_t (
    const int32x8_t & v ) [explicit]

```

Vector of eight int16\_t components

## 5.174 Tellusim::int32x4\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

## Public Types

- enum { **Size** = 4 }

## Public Member Functions

- [int32x4\\_t](#) (const [uint32x4\\_t](#) &v)
- [int32x4\\_t](#) (const [float32x4\\_t](#) &v)
- [int32x4\\_t](#) (const [float64x4\\_t](#) &v)
- [int32x4\\_t](#) (const int32\_t \*v)
- [int32x4\\_t](#) (const int32\_t \*v, int32\_t w)
- [int32x4\\_t](#) (int32\_t v)
- [int32x4\\_t](#) (int32\_t x, int32\_t y, int32\_t z, int32\_t w=0)
- [int16x8\\_t](#) [asi16x8](#) () const

*cast vector data*

- [uint16x8\\_t](#) [asu16x8](#) () const

*cast vector data*

- [uint32x4\\_t](#) [asu32x4](#) () const
- [float16x8\\_t](#) [asf16x8](#) () const
- [float32x4\\_t](#) [asf32x4](#) () const
- void [set](#) (const [int32x4\\_t](#) &v)

*update vector data*

- void [set](#) (int32\_t X, int32\_t Y, int32\_t Z, int32\_t W)
- void [set](#) (const int32\_t \*1 v, int32\_t W)
- void [set](#) (const int32\_t \*1 v)
- void [get](#) (int32\_t \*1 v) const
- template<uint32\_t Index>  
void [set](#) (int32\_t V)
- template<uint32\_t Index>  
int32\_t [get](#) () const
- template<uint32\_t Index>  
[int32x4\\_t](#) [get4](#) () const
- [int32x4\\_t](#) & [operator\\*=](#) (int32\_t v)

*vector to scalar operators*

- [int32x4\\_t](#) & [operator+=](#) (int32\_t v)
- [int32x4\\_t](#) & [operator-=](#) (int32\_t v)
- [int32x4\\_t](#) & [operator &=](#) (int32\_t v)
- [int32x4\\_t](#) & [operator|=](#) (int32\_t v)
- [int32x4\\_t](#) & [operator^=](#) (int32\_t v)
- [int32x4\\_t](#) & [operator<=<=](#) (int32\_t v)
- [int32x4\\_t](#) & [operator>=>=](#) (int32\_t v)
- [int32x4\\_t](#) & [operator\\*=](#) (const [int32x4\\_t](#) &v)

*vector to vector operators*

- [int32x4\\_t](#) & [operator+=](#) (const [int32x4\\_t](#) &v)
- [int32x4\\_t](#) & [operator-=](#) (const [int32x4\\_t](#) &v)
- [int32x4\\_t](#) & [operator &=](#) (const [int32x4\\_t](#) &v)
- [int32x4\\_t](#) & [operator|=](#) (const [int32x4\\_t](#) &v)
- [int32x4\\_t](#) & [operator^=](#) (const [int32x4\\_t](#) &v)
- [int32x4\\_t](#) [zwxxy](#) () const

*swizzle vector*

- [int32x4\\_t](#) [yxwz](#) () const
- int32\_t [sum](#) () const

*sum vector components*

## Public Attributes

- ```
union {  
    struct {  
        int32_t x  
        int32_t y  
        int32_t z  
        int32_t w  
    }  
    int32_t v [Size]  
};
```

## 5.174.1 Detailed Description

Vector of four int32\_t components

## 5.174.2 Constructor &amp; Destructor Documentation

## 5.174.2.1 int32x4\_t()

```
Tellusim::int32x4_t::int32x4_t (  
    const uint32x4_t & v ) [explicit]
```

Vector of four int32\_t components

## 5.175 Tellusim::int32x8\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

## Public Types

- enum { **Size** = 8 }

## Public Member Functions

- [int32x8\\_t](#) (const [uint32x8\\_t](#) &v)
- [int32x8\\_t](#) (const [float32x8\\_t](#) &v)
- [int32x8\\_t](#) (const [float64x8\\_t](#) &v)
- [int32x8\\_t](#) (const int32\_t \*v)
- [int32x8\\_t](#) (int32\_t v)
- [int32x8\\_t](#) (const [int16x8\\_t](#) &v)
- [int32x8\\_t](#) (const [int32x4\\_t](#) &v0, const [int32x4\\_t](#) &v1)
- [int32x8\\_t](#) (int32\_t x0, int32\_t y0, int32\_t z0, int32\_t w0, int32\_t x1, int32\_t y1, int32\_t z1, int32\_t w1)
- [uint32x8\\_t asu32x8](#) () const  
*cast vector data*
- [float32x8\\_t asf32x8](#) () const
- void [set](#) (const [int32x8\\_t](#) &v)  
*update vector data*
- void [set](#) (int32\_t X0, int32\_t Y0, int32\_t Z0, int32\_t W0, int32\_t X1, int32\_t Y1, int32\_t Z1, int32\_t W1)
- void [set](#) (const int32\_t \*1 v)
- void [get](#) (int32\_t \*1 v) const
- template<uint32\_t Index>  
void [set](#) (int32\_t V)
- template<uint32\_t Index>  
int32\_t [get](#) () const
- template<uint32\_t Index>  
[int32x8\\_t get8](#) () const
- [int32x8\\_t & operator\\*=](#) (int32\_t v)  
*vector to scalar operators*
- [int32x8\\_t & operator+=](#) (int32\_t v)
- [int32x8\\_t & operator-=](#) (int32\_t v)
- [int32x8\\_t & operator &=](#) (int32\_t v)
- [int32x8\\_t & operator|=](#) (int32\_t v)
- [int32x8\\_t & operator^=](#) (int32\_t v)
- [int32x8\\_t & operator<=<=](#) (int32\_t v)
- [int32x8\\_t & operator>>=](#) (int32\_t v)
- [int32x8\\_t & operator\\*=](#) (const [int32x8\\_t](#) &v)  
*vector to vector operators*
- [int32x8\\_t & operator+=](#) (const [int32x8\\_t](#) &v)
- [int32x8\\_t & operator-=](#) (const [int32x8\\_t](#) &v)
- [int32x8\\_t & operator &=](#) (const [int32x8\\_t](#) &v)
- [int32x8\\_t & operator|=](#) (const [int32x8\\_t](#) &v)
- [int32x8\\_t & operator^=](#) (const [int32x8\\_t](#) &v)
- [int32x8\\_t xyzw10](#) () const  
*swizzle vector*
- [int32x8\\_t zwxy01](#) () const
- [int32x8\\_t yxwz01](#) () const
- [int32x4\\_t xyzw0](#) () const
- [int32x4\\_t xyzw1](#) () const
- int32\_t [sum](#) () const  
*sum vector components*



## Public Attributes

- ```

union {
    struct {
        int32_t x0
        int32_t y0
        int32_t z0
        int32_t w0
        int32_t x1
        int32_t y1
        int32_t z1
        int32_t w1
    }
    int32_t v [Size]
};

```

## 5.175.1 Detailed Description

Vector of eight int32\_t components

## 5.175.2 Constructor &amp; Destructor Documentation

## 5.175.2.1 int32x8\_t()

```

Tellusim::int32x8_t::int32x8_t (
    const uint32x8_t & v ) [explicit]

```

Vector of eight int32\_t components

## 5.176 Tellusim::IsPtr&lt; Type &gt; Struct Template Reference

```
#include <TellusimBase.h>
```

## Public Types

- enum { **Result** = 0 }

## 5.176.1 Detailed Description

```

template<class Type>
struct Tellusim::IsPtr< Type >

```

Pointer type

### 5.177 Tellusim::RadixMap< Key, Type, Size >::Iterator Class Reference

[Iterator.](#)

```
#include <core/TellusimRadix.h>
```

#### Public Member Functions

- **Iterator** (const [Iterator](#) &it)
- void **clear** ()
- [Iterator](#) & **operator=** (const [Iterator](#) &it)
- **operator bool** () const
- bool **operator==** (const [Iterator](#) &it) const
- bool **operator!=** (const [Iterator](#) &it) const
- [Iterator](#) & **operator++** ()
- [Iterator](#) & **operator--** ()
- [Iterator](#) **operator++** (int32\_t)
- [Iterator](#) **operator--** (int32\_t)
- [Iterator](#) **next** ()
- [Iterator](#) **prev** ()
- Type & **operator\*** ()
- Type \* **operator->** ()
- Type & **get** ()

#### Friends

- class **RadixMap**

#### 5.177.1 Detailed Description

```
template<class Key, class Type, uint32_t Size = 32>
class Tellusim::RadixMap< Key, Type, Size >::Iterator
```

[Iterator.](#)

### 5.178 Tellusim::Json Class Reference

```
#include <format/TellusimJson.h>
```

## Public Member Functions

- **Json** (Type type)
- **Json** (const char \*name, Type type=TypeUnknown)
- **Json** (const [String](#) &name, Type type=TypeUnknown)
- **Json** ([Json](#) \*parent, const char \*name, Type type=TypeUnknown)
- **Json** ([Json](#) \*parent, const [String](#) &name, Type type=TypeUnknown)
- void **clear** ()
  - clear json*
- bool **create** (const char \*str, size\_t size=0, bool owner=false)
  - create json*
- bool **create** (const [String](#) &str, size\_t size=0, bool owner=false)
- bool **load** (const char \*name)
  - load json*
- bool **load** (const [String](#) &name)
- bool **load** ([Stream](#) &stream)
- bool **save** (const char \*name, bool compact=false) const
  - save json*
- bool **save** (const [String](#) &name, bool compact=false) const
- bool **save** ([Stream](#) &stream, bool compact=false) const
- const [Json](#) **getRoot** () const
  - json root*
- [Json](#) **getRoot** ()
- uint32\_t **setParent** ([Json](#) &parent, bool check=true)
  - json parent*
- const [Json](#) **getParent** () const
- [Json](#) **getParent** ()
- [Json](#) **addChild** (const char \*name, Type type=TypeUnknown, bool check=true)
  - json children*
- uint32\_t **addChild** ([Json](#) &child, bool check=true)
- bool **removeChild** ([Json](#) &child)
- void **releaseChildren** ()
- uint32\_t **findChild** (const char \*name) const
- bool **isChild** (const char \*name) const
- const [Json](#) **getChild** (const char \*name) const
- [Json](#) **getChild** (const char \*name)
- uint32\_t **getNumChildren** () const
- const Array< [Json](#) > **getChildren** () const
- Array< [Json](#) > **getChildren** ()
- const [Json](#) **getChild** (uint32\_t index) const
- [Json](#) **getChild** (uint32\_t index)
- [String](#) **getPathName** () const
  - json path name*
- void **setName** (const char \*name)
  - json name*
- void **setName** (const [String](#) &name)
- [String](#) **getName** () const
- void **setType** (Type type)
  - json type*
- Type **getType** () const
- const char \* **getTypeName** () const
- bool **isUnknown** () const
- bool **isNull** () const

- bool **isBool** () const
- bool **isNumber** () const
- bool **isString** () const
- bool **isObject** () const
- bool **isArray** () const
- void **setData** (bool value)
- json data*
- void **setData** (int32\_t value, uint32\_t radix=10)
- void **setData** (uint32\_t value, uint32\_t radix=10)
- void **setData** (uint64\_t value, uint32\_t radix=10)
- void **setData** (float32\_t value, uint32\_t digits=6, bool compact=true, bool exponent=true)
- void **setData** (float64\_t value, uint32\_t digits=12, bool compact=true, bool exponent=true)
- void **setData** (const char \*value)
- void **setData** (const [String](#) &value)
- template<class Type >
- [Json](#) **setData** (const char \*name, Type value, Json::Type type=TypeUnknown)
- [String](#) **getData** () const
- bool **getDataBool** () const
- int32\_t **getDatai32** (uint32\_t radix=10) const
- uint32\_t **getDatau32** (uint32\_t radix=10) const
- uint64\_t **getDatau64** (uint32\_t radix=10) const
- float32\_t **getDataf32** () const
- float64\_t **getDataf64** () const
- [String](#) **getNumber** () const
- [String](#) **getString** () const
- bool **getData** (const char \*name, bool value) const
- int32\_t **getData** (const char \*name, int32\_t value, uint32\_t radix=10) const
- uint32\_t **getData** (const char \*name, uint32\_t value, uint32\_t radix=10) const
- uint64\_t **getData** (const char \*name, uint64\_t value, uint32\_t radix=10) const
- float32\_t **getData** (const char \*name, float32\_t value) const
- float64\_t **getData** (const char \*name, float64\_t value) const
- [String](#) **getData** (const char \*name, const [String](#) &value=[String::null](#)) const
- void **setData** (const char \*\*values, uint32\_t size)
- json array data*
- void **setData** (const [String](#) \*values, uint32\_t size)
- void **setData** (const int32\_t \*values, uint32\_t size, uint32\_t radix=10)
- void **setData** (const uint32\_t \*values, uint32\_t size, uint32\_t radix=10)
- void **setData** (const float32\_t \*values, uint32\_t size, uint32\_t digits=6, bool compact=true, bool exponent=true)
- void **setData** (const float64\_t \*values, uint32\_t size, uint32\_t digits=12, bool compact=true, bool exponent=true)
- template<class Type >
- [Json](#) **setData** (const char \*name, Type \*values, uint32\_t size, Json::Type type=TypeUnknown)
- template<class Type >
- void **setData** (const Array< Type > &values)
- template<class Type >
- void **setData** (const char \*name, const Array< Type > &values)
- uint32\_t **getData** ([String](#) \*values, uint32\_t size) const
- uint32\_t **getData** (int32\_t \*values, uint32\_t size, uint32\_t radix=10) const
- uint32\_t **getData** (uint32\_t \*values, uint32\_t size, uint32\_t radix=10) const
- uint32\_t **getData** (float32\_t \*values, uint32\_t size) const
- uint32\_t **getData** (float64\_t \*values, uint32\_t size) const
- template<class Type >
- uint32\_t **getData** (const char \*name, Type \*values, uint32\_t size) const
- template<class Type >
- uint32\_t **getData** (Array< Type > &values) const
- template<class Type >
- uint32\_t **getData** (const char \*name, Array< Type > &values) const

## Static Public Member Functions

- static const char \* **getTypeName** (Type type)

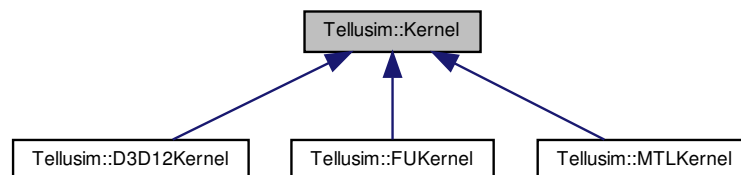
## 5.178.1 Detailed Description

[Json](#)

## 5.179 Tellusim::Kernel Class Reference

```
#include <platform/TellusimKernel.h>
```

Inheritance diagram for Tellusim::Kernel:



## Public Member Functions

- Platform [getPlatform](#) () const  
*kernel platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*kernel device index*
- void [clear](#) ()  
*clear kernel*
- bool [isCreated](#) () const  
*check kernel*
- void [setName](#) (const char \*name)  
*kernel name*
- [String](#) **getName** () const
- bool [create](#) ()  
*create kernel*
- void [setParameters](#) (const [Kernel](#) &kernel)  
*kernel parameters*
- bool **saveState** ([Stream](#) &stream) const
- void [setShader](#) ([Shader](#) &shader, bool owner=false)  
*shader pointer*
- [Shader](#) **getComputeShader** () const
- bool [loadShader](#) (const char \*name, const char \*format,...) 1(3

*load shaders*

- bool bool **loadShaderGLSL** (const char \*name, const char \*format,...) 1(3)
- bool bool bool **loadShader** (const char \*name, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool **loadShaderGLSL** (const char \*name, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool **loadShaderSPIRV** (const char \*name)
- bool [createShader](#) (const char \*src, const char \*format,...) 1(3)

*create shaders*

- bool bool **createShaderGLSL** (const char \*src, const char \*format,...) 1(3)
- bool bool bool **createShader** (const char \*src, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool **createShaderGLSL** (const char \*src, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool **createShaderSPIRV** (const Array< uint32\_t > &data)
- uint32\_t [addSampler](#) ()

*sampler parameters*

- [Kernel](#) & **setSamplers** (uint32\_t num)
- uint32\_t **getNumSamplers** () const
- [Kernel](#) & **setSamplerOffset** (uint32\_t offset)
- uint32\_t **getSamplerOffset** () const
- [Kernel](#) & **setSamplerArray** (uint32\_t index, uint32\_t num, bool array)
- uint32\_t **getSamplerArray** (uint32\_t index) const
- uint32\_t [addTexture](#) ()

*texture parameters*

- [Kernel](#) & **setTextures** (uint32\_t num)
- uint32\_t **getNumTextures** () const
- [Kernel](#) & **setTextureOffset** (uint32\_t offset)
- uint32\_t **getTextureOffset** () const
- [Kernel](#) & **setTextureArray** (uint32\_t index, uint32\_t num, bool array)
- uint32\_t **getTextureArray** (uint32\_t index) const
- uint32\_t [addSurface](#) ()

*surface parameters*

- [Kernel](#) & **setSurfaces** (uint32\_t num)
- uint32\_t **getNumSurfaces** () const
- [Kernel](#) & **setSurfaceOffset** (uint32\_t offset)
- uint32\_t **getSurfaceOffset** () const
- [Kernel](#) & **setSurfaceArray** (uint32\_t index, uint32\_t num, bool array)
- uint32\_t **getSurfaceArray** (uint32\_t index) const
- uint32\_t [addUniform](#) (BindFlags flags=BindFlagNone)

*uniform parameters*

- [Kernel](#) & **setUniforms** (uint32\_t num, BindFlags flags=BindFlagNone)
- uint32\_t **getNumUniforms** () const
- [Kernel](#) & **setUniformOffset** (uint32\_t offset)
- uint32\_t **getUniformOffset** () const
- [Kernel](#) & **setUniformFlags** (uint32\_t index, BindFlags flags)
- BindFlags **getUniformFlags** (uint32\_t index) const
- uint32\_t [addStorage](#) (BindFlags flags=BindFlagNone)

*storage parameters*

- [Kernel](#) & **setStorages** (uint32\_t num, BindFlags flags=BindFlagNone)
- uint32\_t **getNumStorages** () const
- [Kernel](#) & **setStorageOffset** (uint32\_t offset)
- uint32\_t **getStorageOffset** () const
- [Kernel](#) & **setStorageFlags** (uint32\_t index, BindFlags flags)

- BindFlags **getStorageFlags** (uint32\_t index) const
- uint32\_t **addTracing** ()
  - tracing parameters*
- Kernel & **setTracings** (uint32\_t num)
- uint32\_t **getNumTracings** () const
- Kernel & **setTracingOffset** (uint32\_t offset)
- uint32\_t **getTracingOffset** () const
- uint32\_t **addTexel** ()
  - texel parameters*
- Kernel & **setTexels** (uint32\_t num)
- uint32\_t **getNumTexels** () const
- Kernel & **setTexelOffset** (uint32\_t offset)
- uint32\_t **getTexelOffset** () const
- uint32\_t **addTable** (TableType type, uint32\_t size)
  - table parameters*
- uint32\_t **getNumTables** () const
- Kernel & **setTableOffset** (uint32\_t offset)
- uint32\_t **getTableOffset** () const
- Kernel & **setTableType** (uint32\_t index, TableType type, uint32\_t size, BindFlags flags=BindFlagNone)
- TableType **getTableType** (uint32\_t index) const
- uint32\_t **getTableSize** (uint32\_t index) const
- Kernel & **setTableFlags** (uint32\_t index, BindFlags flags)
- BindFlags **getTableFlags** (uint32\_t index) const
- void **setGroupSize** (uint32\_t width, uint32\_t height=1, uint32\_t depth=1)
  - thread group size*
- uint32\_t **getGroupSizeX** () const
- uint32\_t **getGroupSizeY** () const
- uint32\_t **getGroupSizeZ** () const

#### 5.179.1 Detailed Description

[Kernel](#) class

## 5.180 Tellusim::MeshTransform::KeyData< Type > Struct Template Reference

transform data

```
#include <format/TellusimMesh.h>
```

### Public Attributes

- float64\_t **time**
- Type **data**

#### 5.180.1 Detailed Description

```
template<class Type>
```

```
struct Tellusim::MeshTransform::KeyData< Type >
```

transform data

## 5.181 Tellusim::Layer Struct Reference

### Public Member Functions

- **Layer** (uint32\_t base)
- **Layer** (uint32\_t base, uint32\_t size)

### Public Attributes

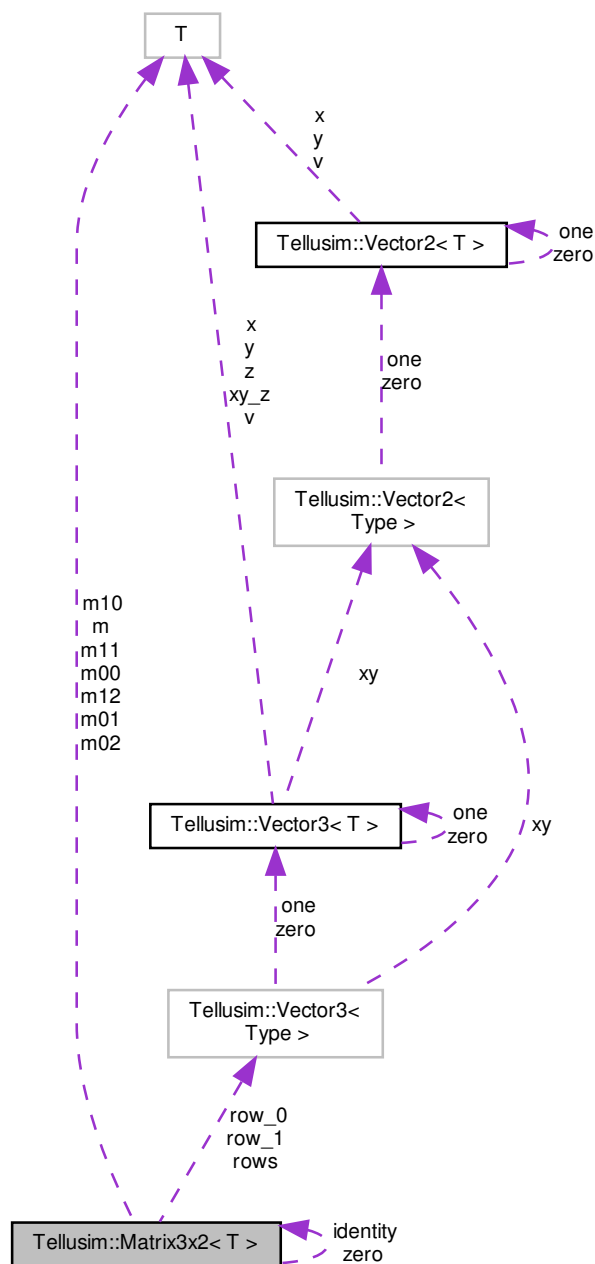
- uint32\_t **base** = 0
- uint32\_t **size** = 1

## 5.182 Tellusim::Matrix3x2< T > Struct Template Reference

```
#include <math/TellusimMatrix.h>
```



Collaboration diagram for Tellusim::Matrix3x2< T >:



### Public Types

- enum {  
    **Rows** = 2,  
    **Columns** = 3,  
    **Size** = Columns \* Rows }
- using **Vector2** = `Tellusim::Vector2< Type >`
- using **Vector3** = `Tellusim::Vector3< Type >`

## Public Member Functions

- **Matrix3x2** (const [Matrix3x2](#) &m)
- **Matrix3x2** (const [Vector3](#) &row\_0, const [Vector3](#) &row\_1)
- **Matrix3x2** (const [Vector2](#) &col\_0, const [Vector2](#) &col\_1, const [Vector2](#) &col\_2)
- **Matrix3x2** (Type m00, Type m01, Type m02, Type m10, Type m11, Type m12)
- **Matrix3x2** (const Type \*1 m, uint32\_t size=[Size](#), bool row\_major=true)
- template<class CType >  
**Matrix3x2** (const [Tellusim::Matrix3x2](#)< CType > &m)
- **Matrix3x2** (Type v)
- void **set** (const [Vector3](#) &r0, const [Vector3](#) &r1)  
*update matrix data*
- void **set** (const [Vector3](#) &col\_0, const [Vector3](#) &col\_1, const [Vector3](#) &col\_2)
- void **set** (const Type \*1 m, uint32\_t size=[Size](#), bool row\_major=true)
- void **get** (Type \*1 m, uint32\_t size=[Size](#), bool row\_major=true) const
- [Matrix3x2](#) & **operator\*=** (const [Matrix3x2](#) &m1)  
*matrix to matrix multiplication*
- void **setZero** ()  
*zero matrix*
- bool **isZero** () const
- void **setIdentity** ()  
*identity matrix*
- bool **isIdentity** () const
- void **setScale** (const [Vector2](#) &s)  
*scaling matrix*
- void **setScale** (Type x, Type y)
- [Vector2](#) **getScale** () const
- void **setTranslate** (const [Vector2](#) &t)  
*translation matrix*
- void **setTranslate** (Type x, Type y)
- [Vector2](#) **getTranslate** () const
- void **setRotate** (Type angle)  
*rotation matrix*
- [Matrix3x2](#) **getRotate** () const
- Type **getDeterminant** () const  
*matrix determinant*
- void **setRow** (uint32\_t index, const [Vector3](#) &r)  
*matrix rows*
- const [Vector3](#) & **getRow** (uint32\_t index) const
- [Vector3](#) & **getRow** (uint32\_t index)
- void **setColumn** (uint32\_t index, const [Vector2](#) &c)  
*matrix columns*
- [Vector2](#) **getColumn** (uint32\_t index) const
- const [Vector3](#) & **operator[]** (uint32\_t index) const  
*matrix data*
- [Vector3](#) & **operator[]** (uint32\_t index)

## Static Public Member Functions

- static [Matrix3x2](#) **scale** (const [Vector2](#) &s)
- static [Matrix3x2](#) **scale** (Type x, Type y)
- static [Matrix3x2](#) **scale** (Type s)
- static [Matrix3x2](#) **translate** (const [Vector2](#) &t)
- static [Matrix3x2](#) **translate** (Type x, Type y)
- static [Matrix3x2](#) **rotate** (Type angle)

## Public Attributes

- ```

union {
    struct {
        Type m00
        Type m01
        Type m02
        Type m10
        Type m11
        Type m12
    }
    struct {
        Vector3 row_0
        Vector3 row_1
    }
    Vector3 rows [Rows]
    Type m [Size]
};

```

## Static Public Attributes

- static const [Matrix3x2 zero](#)  
*default matrices*
- static const [Matrix3x2 identity](#)

## 5.182.1 Detailed Description

```

template<class T>
struct Tellusim::Matrix3x2< T >

```

[Matrix3x2](#) class

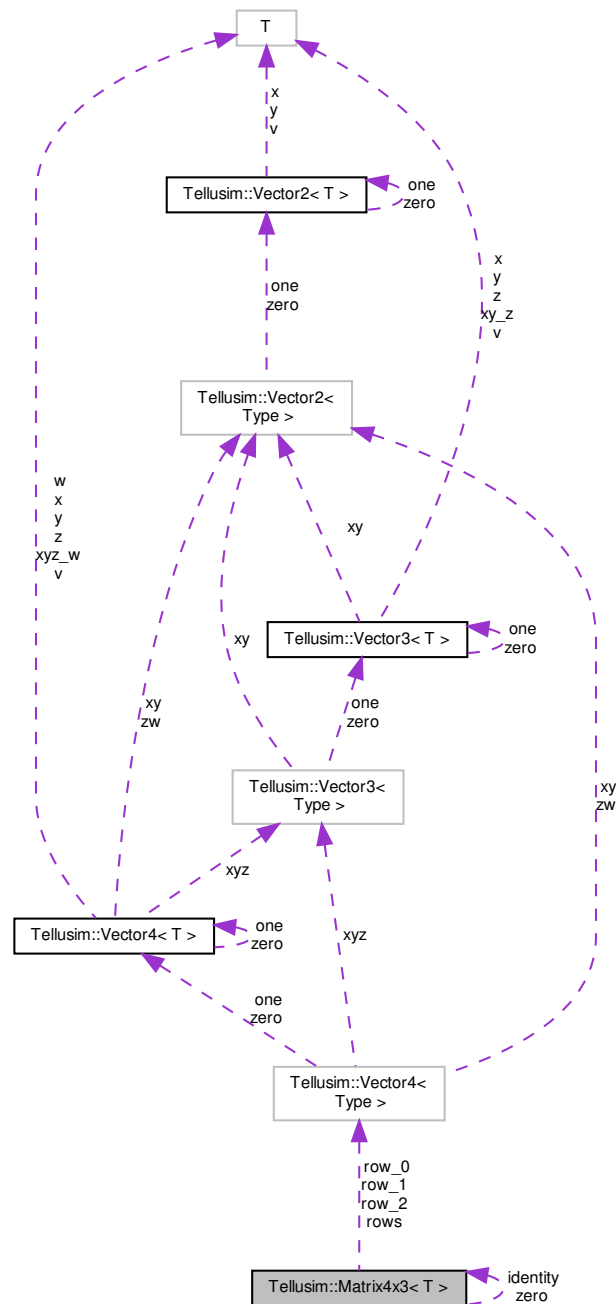
## 5.183 Tellusim::Matrix4x3&lt; T &gt; Struct Template Reference

```

#include <math/TellusimMatrix.h>

```

Collaboration diagram for `Tellusim::Matrix4x3< T >`:



#### Public Types

- enum {  
**Rows** = 3,  
**Columns** = 4,  
**Size** = Columns \* Rows }
- using **Vector3** = `Tellusim::Vector3< Type >`

- using **Vector4** = [Tellusim::Vector4](#)< Type >
- using **Matrix4x4** = [Tellusim::Matrix4x4](#)< Type >
- using **Quaternion** = [Tellusim::Quaternion](#)< Type >

#### Public Member Functions

- **Matrix4x3** (const [Matrix4x3](#) &m)
- **Matrix4x3** (const [Vector4](#) &row\_0, const [Vector4](#) &row\_1, const [Vector4](#) &row\_2)
- **Matrix4x3** (const [Vector3](#) &col\_0, const [Vector3](#) &col\_1, const [Vector3](#) &col\_2, const [Vector3](#) &col\_3)
- **Matrix4x3** (Type m00, Type m01, Type m02, Type m03, Type m10, Type m11, Type m12, Type m13, Type m20, Type m21, Type m22, Type m23)
- **Matrix4x3** (const [Matrix4x4](#) &m)
- **Matrix4x3** (const [Quaternion](#) &q)
- **Matrix4x3** (const Type \*1 m, uint32\_t size=[Size](#), bool row\_major=true)
- template<class CType >  
**Matrix4x3** (const [Tellusim::Matrix4x3](#)< CType > &m)
- template<class CType >  
**Matrix4x3** (const [Tellusim::Matrix4x4](#)< CType > &m)
- **Matrix4x3** (Type v)
- void **set** (const [Vector4](#) &r0, const [Vector4](#) &r1, const [Vector4](#) &r2)  
*update matrix data*
- void **set** (const [Vector3](#) &col\_0, const [Vector3](#) &col\_1, const [Vector3](#) &col\_2, const [Vector3](#) &col\_3)
- void **set** (const Type \*1 m, uint32\_t size=[Size](#), bool row\_major=true)
- void **get** (Type \*1 m, uint32\_t size=[Size](#), bool row\_major=true) const
- [Matrix4x3](#) & **operator\*=** (const [Matrix4x3](#)< Type > &m1)  
*matrix to matrix multiplication*
- void **setZero** ()  
*zero matrix*
- bool **isZero** () const
- void **setIdentity** ()  
*identity matrix*
- bool **isIdentity** () const
- void **setScale** (const [Vector3](#) &s)  
*scaling matrix*
- void **setScale** (Type x, Type y, Type z)
- [Vector3](#) **getScale** () const
- void **setTranslate** (const [Vector3](#) &t)  
*translation matrix*
- void **setTranslate** (Type x, Type y, Type z)
- [Vector3](#) **getTranslate** () const
- void **setRotateX** (Type angle)  
*rotation matrix*
- void **setRotateY** (Type angle)
- void **setRotateZ** (Type angle)
- void **setRotate** (const [Vector3](#) &axis, Type angle)
- void **setRotate** (Type x, Type y, Type z, Type angle)
- [Matrix4x3](#) **getRotate** () const
- void **setLookAt** (const [Vector3](#) &from, const [Vector3](#) &to, const [Vector3](#) &up)  
*look at matrix*
- void **setPlaceTo** (const [Vector3](#) &to, const [Vector3](#) &from, const [Vector3](#) &up)  
*place to matrix*
- void **setCubeAt** (const [Vector3](#) &from, uint32\_t face)

- cube at matrix*
- void **setBasis** (const **Vector3** &normal, const **Vector3** &t)
- right-handed orthonormal basis*
- Type **getDeterminant** () const
- matrix determinant*
- void **setComponents** (const **Vector3** &t, const **Quaternion** &r)
- matrix composition*
- void **setComponents** (const **Vector3** &t, const **Quaternion** &r, const **Vector3** &s)
- void **getComponents** (**Vector3** &t, **Quaternion** &r) const
- void **getComponents** (**Vector3** &t, **Quaternion** &r, **Vector3** &s) const
- void **setRow** (uint32\_t index, const **Vector4** &r)
- matrix rows*
- const **Vector4** & **getRow** (uint32\_t index) const
- **Vector4** & **getRow** (uint32\_t index)
- void **setColumn** (uint32\_t index, const **Vector3** &c)
- matrix columns*
- **Vector3** **getColumn** (uint32\_t index) const
- const **Vector4** & **operator[]** (uint32\_t index) const
- matrix data*
- **Vector4** & **operator[]** (uint32\_t index)

#### Static Public Member Functions

- static **Matrix4x3** **scale** (const **Vector3** &s)
- static **Matrix4x3** **scale** (Type x, Type y, Type z)
- static **Matrix4x3** **scale** (Type s)
- static **Matrix4x3** **translate** (const **Vector3** &t)
- static **Matrix4x3** **translate** (Type x, Type y, Type z)
- static **Matrix4x3** **rotateX** (Type angle)
- static **Matrix4x3** **rotateY** (Type angle)
- static **Matrix4x3** **rotateZ** (Type angle)
- static **Matrix4x3** **rotate** (const **Vector3** &axis, Type angle)
- static **Matrix4x3** **rotate** (Type x, Type y, Type z, Type angle)
- static **Matrix4x3** **lookAt** (const **Vector3** &from, const **Vector3** &to, const **Vector3** &up)
- static **Matrix4x3** **placeTo** (const **Vector3** &to, const **Vector3** &from, const **Vector3** &up)
- static **Matrix4x3** **cubeAt** (const **Vector3** &from, uint32\_t face)
- static **Matrix4x3** **basis** (const **Vector3** &normal)
- static **Matrix4x3** **basis** (const **Vector3** &normal, const **Vector3** &t)
- static **Matrix4x3** **compose** (const **Vector3** &t, const **Quaternion** &r)
- static **Matrix4x3** **compose** (const **Vector3** &t, const **Quaternion** &r, const **Vector3** &s)

#### Public Attributes

- - union {
    - struct {
 Type **m00**
 Type **m01**
 Type **m02**
 Type **m03**
 Type **m10**
 Type **m11**

```

    Type m12
    Type m13
    Type m20
    Type m21
    Type m22
    Type m23
}
struct {
    Vector4 row_0
    Vector4 row_1
    Vector4 row_2
}
Vector4 rows [Rows]
Type m [Size]
};

```

#### Static Public Attributes

- static const [Matrix4x3 zero](#)  
*default matrices*
- static const [Matrix4x3 identity](#)

#### 5.183.1 Detailed Description

```

template<class T>
struct Tellusim::Matrix4x3< T >

```

[Matrix4x3](#) class

## 5.184 Tellusim::Matrix4x4< T > Struct Template Reference

```

#include <math/TellusimMatrix.h>

```





- using **Vector4** = [Tellusim::Vector4](#)< Type >
- using **Matrix4x3** = [Tellusim::Matrix4x3](#)< Type >
- using **Quaternion** = [Tellusim::Quaternion](#)< Type >

#### Public Member Functions

- **Matrix4x4** (const [Matrix4x4](#) &m)
- **Matrix4x4** (const [Vector4](#) &row\_0, const [Vector4](#) &row\_1, const [Vector4](#) &row\_2, const [Vector4](#) &row\_3)
- **Matrix4x4** (Type m00, Type m01, Type m02, Type m03, Type m10, Type m11, Type m12, Type m13, Type m20, Type m21, Type m22, Type m23, Type m30, Type m31, Type m32, Type m33)
- **Matrix4x4** (const [Matrix4x3](#) &m)
- **Matrix4x4** (const [Quaternion](#) &q)
- **Matrix4x4** (const Type \*1 m, uint32\_t size=[Size](#), bool row\_major=true)
- template<class CType >  
**Matrix4x4** (const [Tellusim::Matrix4x3](#)< CType > &m)
- template<class CType >  
**Matrix4x4** (const [Tellusim::Matrix4x4](#)< CType > &m)
- **Matrix4x4** (Type v)
- void **set** (const [Vector4](#) &r0, const [Vector4](#) &r1, const [Vector4](#) &r2, const [Vector4](#) &r3, bool row\_major=true)  
*update matrix data*
- void **set** (const Type \*1 m, uint32\_t size=[Size](#), bool row\_major=true)
- void **get** (Type \*1 m, uint32\_t size=[Size](#), bool row\_major=true) const
- [Matrix4x4](#) & **operator\*=** (const [Matrix4x4](#)< Type > &m1)  
*matrix to matrix multiplication*
- void **setZero** ()  
*zero matrix*
- bool **isZero** () const
- void **setIdentity** ()  
*identity matrix*
- bool **isIdentity** () const
- void **setScale** (const [Vector3](#) &s)  
*scaling matrix*
- void **setScale** (Type x, Type y, Type z)
- [Vector3](#) **getScale** () const
- void **setTranslate** (const [Vector3](#) &t)  
*translation matrix*
- void **setTranslate** (Type x, Type y, Type z)
- [Vector3](#) **getTranslate** () const
- void **setRotateX** (Type angle)  
*rotation matrix*
- void **setRotateY** (Type angle)
- void **setRotateZ** (Type angle)
- void **setRotate** (const [Vector3](#) &axis, Type angle)
- void **setRotate** (Type x, Type y, Type z, Type angle)
- [Matrix4x4](#) **getRotate** () const
- void **setLookAt** (const [Vector3](#) &from, const [Vector3](#) &to, const [Vector3](#) &up)  
*look at matrix*
- void **setPlaceTo** (const [Vector3](#) &from, const [Vector3](#) &to, const [Vector3](#) &up)  
*place to matrix*
- void **setCubeAt** (const [Vector3](#) &from, uint32\_t face)  
*cube at matrix*
- void **setBasis** (const [Vector3](#) &normal, const [Vector3](#) &t)

- right-handed orthonormal basis*
- Type `getDeterminant` () const
- matrix determinant*
- void `setComponents` (const `Vector3` &t, const `Quaternion` &r, const `Vector3` &s)
- matrix composition*
- void `getComponents` (`Vector3` &t, `Quaternion` &r, `Vector3` &s) const
- void `setOrtho` (Type left, Type right, Type bottom, Type top, Type znear, Type zfar)
- ortho matrix*
- void `setFrustum` (Type left, Type right, Type bottom, Type top, Type znear, Type zfar, bool reverse=false)
- frustum matrix*
- void `setFrustum` (Type left, Type right, Type bottom, Type top, Type znear, bool reverse=false)
- infinite frustum matrix*
- void `setPerspective` (Type fov, Type aspect, Type znear, Type zfar, bool reverse=false)
- perspective matrix*
- void `setPerspective` (Type fov, Type aspect, Type znear, bool reverse=false)
- infinite perspective matrix*
- void `setRow` (uint32\_t index, const `Vector4` &r)
- matrix rows*
- const `Vector4` & `getRow` (uint32\_t index) const
- `Vector4` & `getRow` (uint32\_t index)
- void `setColumn` (uint32\_t index, const `Vector4` &c)
- matrix columns*
- `Vector4` `getColumn` (uint32\_t index) const
- const `Vector4` & `operator[]` (uint32\_t index) const
- matrix data*
- `Vector4` & `operator[]` (uint32\_t index)

#### Static Public Member Functions

- static `Matrix4x4` `scale` (const `Vector3` &s)
- static `Matrix4x4` `scale` (Type x, Type y, Type z)
- static `Matrix4x4` `scale` (Type s)
- static `Matrix4x4` `translate` (const `Vector3` &t)
- static `Matrix4x4` `translate` (Type x, Type y, Type z)
- static `Matrix4x4` `rotateX` (Type angle)
- static `Matrix4x4` `rotateY` (Type angle)
- static `Matrix4x4` `rotateZ` (Type angle)
- static `Matrix4x4` `rotate` (const `Vector3` &axis, Type angle)
- static `Matrix4x4` `rotate` (Type x, Type y, Type z, Type angle)
- static `Matrix4x4` `lookAt` (const `Vector3` &from, const `Vector3` &to, const `Vector3` &up)
- static `Matrix4x4` `placeTo` (const `Vector3` &from, const `Vector3` &to, const `Vector3` &up)
- static `Matrix4x4` `cubeAt` (const `Vector3` &from, uint32\_t face)
- static `Matrix4x4` `basis` (const `Vector3` &normal)
- static `Matrix4x4` `basis` (const `Vector3` &normal, const `Vector3` &t)
- static `Matrix4x4` `compose` (const `Vector3` &t, const `Quaternion` &r, const `Vector3` &s)
- static `Matrix4x4` `ortho` (Type left, Type right, Type bottom, Type top, Type znear, Type zfar)
- static `Matrix4x4` `frustum` (Type left, Type right, Type bottom, Type top, Type znear, Type zfar, bool reverse=false)
- static `Matrix4x4` `frustum` (Type left, Type right, Type bottom, Type top, Type znear, bool reverse=false)
- static `Matrix4x4` `perspective` (Type fov, Type aspect, Type znear, Type zfar, bool reverse=false)
- static `Matrix4x4` `perspective` (Type fov, Type aspect, Type znear, bool reverse=false)

## Public Attributes

- ```

union {
    struct {
        Type m00
        Type m01
        Type m02
        Type m03
        Type m10
        Type m11
        Type m12
        Type m13
        Type m20
        Type m21
        Type m22
        Type m23
        Type m30
        Type m31
        Type m32
        Type m33
    }
    struct {
        Vector4 row_0
        Vector4 row_1
        Vector4 row_2
        Vector4 row_3
    }
    Vector4 rows [Rows]
    Type m [Size]
};

```

## Static Public Attributes

- static const [Matrix4x4 zero](#)  
*default matrices*
- static const [Matrix4x4 identity](#)

## 5.184.1 Detailed Description

```

template<class T>
struct Tellusim::Matrix4x4< T >

```

[Matrix4x4](#) class

## 5.185 Tellusim::MatrixNxM&lt; Type, N, M &gt; Struct Template Reference

```

#include <math/TellusimNumerical.h>

```

## Public Types

- enum {  
    **Rows** = M,  
    **Columns** = N,  
    **Size** = Rows \* Columns }
- using **VectorN** = Tellusim::VectorN< Type, N >
- using **VectorM** = Tellusim::VectorN< Type, M >
- using **VectorNxM** = Tellusim::VectorN< Type, N \*M >

## Public Member Functions

- **MatrixNxM** (const **MatrixNxM** &matrix)
- **MatrixNxM** (const Type &value)
- **MatrixNxM** (const Type \*matrix)
- **MatrixNxM** (const **VectorNxM** &vector)
- **MatrixNxM** (const InitializerList< Type > &list)
- **MatrixNxM** (const InitializerList< **VectorN** > &list)
- template<class CType >  
    **MatrixNxM** (const **MatrixNxM**< CType, N, M > &matrix)
- void **set** (const Type &value)  
    *update matrix data*
- void **set** (const Type \*1 matrix)
- void **set** (const **MatrixNxM** &matrix)
- void **set** (const **VectorNxM** &vector)
- void **set** (const InitializerList< Type > &list)
- void **set** (const InitializerList< **VectorN** > &list)
- void **get** (Type \*1 matrix)
- void **setZero** ()  
    *zero matrix*
- void **setIdentity** ()  
    *identity matrix*
- **MatrixNxM** & **operator\*=** (const Type &value)  
    *matrix to scalar operators*
- **MatrixNxM** & **operator/=** (const Type &value)
- **MatrixNxM** & **operator=** (const **MatrixNxM** &matrix)  
    *matrix to matrix operators*
- void **setRow** (uint32\_t index, const **VectorN** &r)  
    *matrix rows*
- const **VectorN** & **getRow** (uint32\_t index) const
- **VectorN** & **getRow** (uint32\_t index)
- void **setColumn** (uint32\_t index, const **VectorM** &c)  
    *matrix columns*
- **VectorM** **getColumn** (uint32\_t index) const
- const **VectorN** & **operator[]** (uint32\_t index) const  
    *matrix data*
- **VectorN** & **operator[]** (uint32\_t index)

## Public Attributes

- - union {
    - [VectorN](#) rows [Rows]
    - Type m [Size]

};

## 5.185.1 Detailed Description

```
template<class Type, uint32_t N, uint32_t M>
struct Tellusim::MatrixNxM< Type, N, M >
```

[MatrixNxM](#) class

## 5.186 Tellusim::Mesh Class Reference

```
#include <format/TellusimMesh.h>
```

## Public Types

- enum [Flags](#) {
  - FlagNone** = 0,
  - FlagEmbed** = (1 << 0),
  - Flag32Bit** = (1 << 1) }

*[Mesh](#) flags.*

- enum [Basis](#) {
  - BasisUnknown** = 0,
  - BasisXUpRight**,
  - BasisYUpRight**,
  - BasisZUpRight**,
  - BasisXUpLeft**,
  - BasisYUpLeft**,
  - BasisZUpLeft**,
  - BasisZUpMaya**,
  - NumBases** }

*mesh basis*

- enum **Axis** {
  - AxisUnknown** = 0,
  - AxisPX**,
  - AxisPY**,
  - AxisPZ**,
  - AxisNX**,
  - AxisNY**,
  - AxisNZ**,
  - NumAxes** }

## Public Member Functions

- void **clear** ()  
*clear mesh*
- bool **isLoading** () const  
*check mesh*
- bool **info** (const char \*name, **Flags** flags=FlagNone, **Async** \*async=nullptr)  
*info mesh*
- bool **info** (const **String** &name, **Flags** flags=FlagNone, **Async** \*async=nullptr)
- bool **info** (**Stream** &stream, **Flags** flags=FlagNone, **Async** \*async=nullptr)
- bool **info** (const char \*name, **Async** \*async)
- bool **info** (const **String** &name, **Async** \*async)
- bool **info** (**Stream** &stream, **Async** \*async)
- bool **load** (const char \*name, **Flags** flags=FlagNone, **Async** \*async=nullptr)  
*load mesh*
- bool **load** (const **String** &name, **Flags** flags=FlagNone, **Async** \*async=nullptr)
- bool **load** (**Stream** &stream, **Flags** flags=FlagNone, **Async** \*async=nullptr)
- bool **load** (const char \*name, **Async** \*async)
- bool **load** (const **String** &name, **Async** \*async)
- bool **load** (**Stream** &stream, **Async** \*async)
- bool **save** (const char \*name, **Flags** flags=FlagNone) const  
*save mesh*
- bool **save** (const **String** &name, **Flags** flags=FlagNone) const
- bool **save** (**Stream** &stream, **Flags** flags=FlagNone) const
- void **setName** (const char \*name)  
*mesh name*
- **String** **getName** () const
- bool **setBasis** (Axis front, Axis right, Axis up)
- bool **setBasis** (**Basis** basis)
- Axis **getFrontAxis** () const
- Axis **getRightAxis** () const
- Axis **getUpAxis** () const
- **Basis** **getBasis** () const
- const char \* **getFrontAxisName** () const
- const char \* **getRightAxisName** () const
- const char \* **getUpAxisName** () const
- **String** **getBasisName** () const
- void **clearNodes** ()  
*mesh nodes*
- void **reserveNodes** (uint32\_t num\_nodes)
- uint32\_t **addNode** (**MeshNode** &node, bool check=true)
- bool **removeNode** (**MeshNode** &node)
- uint32\_t **findNode** (const **MeshNode** &node) const
- uint32\_t **findNode** (const char \*name) const
- uint32\_t **getNumNodes** () const
- const Array< **MeshNode** > **getNodes** () const
- Array< **MeshNode** > **getNodes** ()
- const **MeshNode** **getNode** (uint32\_t index) const
- **MeshNode** **getNode** (uint32\_t index)
- void **createLocalTransforms** (const **Matrix4x3d** &itransform=Matrix4x3d::identity)
- void **createGlobalTransforms** (const **Matrix4x3d** &transform=Matrix4x3d::identity)
- void **clearGeometries** ()  
*mesh geometries*

- void **reserveGeometries** (uint32\_t num\_geometries)
- uint32\_t **addGeometry** ([MeshGeometry](#) &geometry, bool check=true)
- uint32\_t **addGeometry** ([MeshGeometry](#) &geometry, [MeshNode](#) &node, bool check=true)
- bool **removeGeometry** ([MeshGeometry](#) &geometry)
- bool **replaceGeometry** ([MeshGeometry](#) &old\_geometry, [MeshGeometry](#) &geometry)
- uint32\_t **findGeometry** (const [MeshGeometry](#) &geometry) const
- uint32\_t **findGeometry** (const char \*name) const
- uint32\_t **getNumGeometries** () const
- const Array< [MeshGeometry](#) > **getGeometries** () const
- Array< [MeshGeometry](#) > **getGeometries** ()
- const [MeshGeometry](#) **getGeometry** (uint32\_t index) const
- [MeshGeometry](#) **getGeometry** (uint32\_t index)
- bool **hasGeometryIndices** (MeshIndices::Type type) const
- bool **hasGeometryAttribute** (MeshAttribute::Type type) const
- size\_t **getNumGeometryPositions** () const
- size\_t **getNumGeometryPrimitives** () const
- void **clearAnimations** ()
- mesh animations*
- void **reserveAnimations** (uint32\_t num\_animations)
- uint32\_t **addAnimation** ([MeshAnimation](#) &animation, bool check=true)
- bool **removeAnimation** ([MeshAnimation](#) &animation)
- bool **replaceAnimation** ([MeshAnimation](#) &old\_animation, [MeshAnimation](#) &animation)
- uint32\_t **findAnimation** (const [MeshAnimation](#) &animation) const
- uint32\_t **findAnimation** (const char \*name) const
- uint32\_t **getNumAnimations** () const
- const Array< [MeshAnimation](#) > **getAnimations** () const
- Array< [MeshAnimation](#) > **getAnimations** ()
- const [MeshAnimation](#) **getAnimation** (uint32\_t index) const
- [MeshAnimation](#) **getAnimation** (uint32\_t index)
- [BoundingBoxd](#) **getBoundingBox** () const
- mesh bound box*
- [BoundingBoxd](#) **getBoundingBox** (const [MeshNode](#) &node) const
- [BoundingBoxd](#) **getBoundingBox** (const [MeshGeometry](#) &geometry) const
- bool **createBounds** (bool force, [Async](#) \*async=nullptr)
- bool **createBounds** ([Async](#) \*async=nullptr)
- bool **createBasis** (bool force, [Async](#) \*async=nullptr)
- bool **createBasis** (float32\_t angle, bool force, [Async](#) \*async=nullptr)
- bool **createBasis** ([Async](#) \*async=nullptr)
- bool **createBasis** (float32\_t angle, [Async](#) \*async=nullptr)
- bool **createNormals** (bool force, [Async](#) \*async=nullptr)
- bool **createNormals** (float32\_t angle, bool force, [Async](#) \*async=nullptr)
- bool **createNormals** ([Async](#) \*async=nullptr)
- bool **createNormals** (float32\_t angle, [Async](#) \*async=nullptr)
- bool **createTangents** (bool force, [Async](#) \*async=nullptr)
- bool **createTangents** ([Async](#) \*async=nullptr)
- bool **createIslands** (uint32\_t max\_attributes, uint32\_t max\_primitives, bool force, [Async](#) \*async=nullptr)
- bool **createIslands** (uint32\_t max\_attributes, uint32\_t max\_primitives, [Async](#) \*async=nullptr)
- bool **optimizeIndices** (uint32\_t cache, bool transparent, [Async](#) \*async=nullptr)
- optimize indices order for cache*
- bool **optimizeIndices** ([Async](#) \*async=nullptr)
- bool **optimizeIndices** (uint32\_t cache, [Async](#) \*async=nullptr)
- bool **optimizeAttributes** ([Async](#) \*async=nullptr)
- optimize attributes and make single indices.*
- void **optimizeMaterials** ()

- optimize materials remove duplicates*
- bool [optimizeWinding](#) (bool clockwise=false)
- void [optimizeGeometries](#) (float32\_t threshold=1e-3f, uint32\_t depth=16)
- void [optimizeAnimations](#) (float32\_t threshold=1e-6f)
- bool [optimizeOrder](#) ()
- optimize node and geometry order*
- void [mergeGeometries](#) ()
- merge geometries*
- bool [packAttributes](#) (bool remove=true)
- pack attributes*
- bool **unpackAttributes** (bool remove=true)
- bool [setTransform](#) (const [Vector3d](#) &scale)
- apply transform*
- size\_t [getMemory](#) () const
- memory usage*

#### 5.186.1 Detailed Description

[Mesh](#) class

#### 5.186.2 Member Function Documentation

##### 5.186.2.1 [createBounds\(\)](#)

```
bool Tellusim::Mesh::createBounds (
    bool force,
    Async * async = nullptr )
```

create bounds

##### Parameters

|              |                          |
|--------------|--------------------------|
| <i>force</i> | Force bounding creation. |
|--------------|--------------------------|

##### 5.186.2.2 [createBasis\(\)](#)

```
bool Tellusim::Mesh::createBasis (
    bool force,
    Async * async = nullptr )
```

create tangent basis

##### Parameters

|              |                       |
|--------------|-----------------------|
| <i>force</i> | Force basis creation. |
|--------------|-----------------------|



### 5.186.2.3 createNormals()

```
bool Tellusim::Mesh::createNormals (
    bool force,
    Async * async = nullptr )
```

create normals

#### Parameters

|              |                             |
|--------------|-----------------------------|
| <i>force</i> | Force normals creation.     |
| <i>angle</i> | Smoothing angle in degrees. |

### 5.186.2.4 createTangents()

```
bool Tellusim::Mesh::createTangents (
    bool force,
    Async * async = nullptr )
```

create tangents

#### Parameters

|              |                          |
|--------------|--------------------------|
| <i>force</i> | Force tangents creation. |
|--------------|--------------------------|

### 5.186.2.5 createIslands()

```
bool Tellusim::Mesh::createIslands (
    uint32_t max_attributes,
    uint32_t max_primitives,
    bool force,
    Async * async = nullptr )
```

create islands

#### Parameters

|                       |                                          |
|-----------------------|------------------------------------------|
| <i>max_attributes</i> | Maximum number of attributes per island. |
| <i>max_primitives</i> | Maximum number of primitives per island. |
| <i>force</i>          | Force islands creation.                  |

### 5.186.2.6 optimizeWinding()

```
bool Tellusim::Mesh::optimizeWinding (
```

```
bool clockwise = false )
```

optimize winding based on node transforms

#### Parameters

|                  |                                 |
|------------------|---------------------------------|
| <i>clockwise</i> | Optimize for clockwise winding. |
|------------------|---------------------------------|

#### 5.186.2.7 optimizeGeometries()

```
void Tellusim::Mesh::optimizeGeometries (
    float32_t threshold = 1e-3f,
    uint32_t depth = 16 )
```

optimize geometries remove duplicates

#### Parameters

|                   |                                                   |
|-------------------|---------------------------------------------------|
| <i>threshold</i>  | <a href="#">Spatial</a> compare threshold.        |
| <i>depth</i>      | Number of geometries to compare.                  |
| <i>step</i>       | Geometry bound optimization step.                 |
| <i>mask</i>       | Geometry bound optimization axis mask.            |
| <i>iterations</i> | Number of geometry bound optimization iterations. |

#### 5.186.2.8 optimizeAnimations()

```
void Tellusim::Mesh::optimizeAnimations (
    float32_t threshold = 1e-6f )
```

optimize animation transforms

#### Parameters

|                  |                    |
|------------------|--------------------|
| <i>threshold</i> | Compare threshold. |
|------------------|--------------------|

### 5.187 Tellusim::MeshAnimation Class Reference

```
#include <format/TellusimMesh.h>
```

#### Public Member Functions

- **MeshAnimation** (const char \*name=nullptr)
- **MeshAnimation** ([Mesh](#) &mesh, const char \*name=nullptr)
- void [clear](#) ()

- clear animation*
- uint32\_t **getIndex** () const
- animation index*
- void **setName** (const char \*name)
- animation name*
- String **getName** () const
- void **setMesh** (Mesh &mesh, bool check=true)
- animation mesh*
- const Mesh **getMesh** () const
- Mesh **getMesh** ()
- float64\_t **getMinTime** () const
- animation range*
- float64\_t **getMaxTime** () const
- void **setNumTransforms** (uint32\_t num\_transforms)
- animation transforms*
- uint32\_t **getNumTransforms** () const
- const Array< MeshTransform > **getTransforms** () const
- Array< MeshTransform > **getTransforms** ()
- const MeshTransform **getTransform** (uint32\_t node) const
- MeshTransform **getTransform** (uint32\_t node)
- void **setTransform** (float64\_t time, uint32\_t node, const Matrix4x3d &transform, float32\_t threshold=1e-6f)
- void **setTranslate** (float64\_t time, uint32\_t node, const Vector3d &translate, float32\_t threshold=1e-6f)
- void **setRotate** (float64\_t time, uint32\_t node, const Quaternionf &rotate, float32\_t threshold=1e-6f)
- void **setScale** (float64\_t time, uint32\_t node, const Vector3f &scale, float32\_t threshold=1e-6f)
- void **setMorph** (float64\_t time, uint32\_t node, const Vector4f &morph, float32\_t threshold=1e-6f)
- void **setTime** (float64\_t time, const Matrix4x3d &transform=Matrix4x3d::identity, bool loop=true, float64\_t from=-Maxf32, float64\_t to=Maxf32)
- animation transform*
- void **setTime** (float64\_t time, bool loop, float64\_t from=-Maxf32, float64\_t to=Maxf32)
- const Matrix4x3d & **getLocalTransform** (uint32\_t node) const
- const Matrix4x3d & **getLocalTransform** (const MeshNode &node) const
- const Matrix4x3d & **getLocalTransform** (const MeshJoint &joint) const
- const Matrix4x3d & **getGlobalTransform** (uint32\_t node) const
- const Matrix4x3d & **getGlobalTransform** (const MeshNode &node) const
- const Matrix4x3d & **getGlobalTransform** (const MeshJoint &joint) const
- const Vector4f & **getMorphTransform** (uint32\_t node) const
- const Vector4f & **getMorphTransform** (const MeshNode &node) const
- BoundingBoxd **getBoundingBox** () const
- animation bound box*
- BoundingBoxd **getBoundingBox** (const MeshNode &node) const
- BoundingBoxd **getBoundingBox** (const MeshGeometry &geometry) const
- BoundingBoxd **getBoundingBox** (const MeshGeometry &geometry, const Vector4f &morph) const
- void **setTransform** (const Vector3d &scale)
- apply transform*
- void **setTransform** (const Matrix4x3d &transform)
- void **optimizeTransforms** (float32\_t threshold=1e-6f)
- optimize animation*
- size\_t **getMemory** () const
- memory usage*

## Friends

- class **Mesh**

## 5.187.1 Detailed Description

[MeshAnimation](#) class

## 5.188 Tellusim::MeshAttachment Class Reference

```
#include <format/TellusimMesh.h>
```

## Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagBool** = (1 << 0),  
**FlagScalarf32** = (1 << 1),  
**FlagVector4f** = (1 << 2),  
**FlagColor** = (1 << 3),  
**FlagName** = (1 << 4) }

*Attachment flags.*

## Public Member Functions

- **MeshAttachment** (const char \*name=nullptr)
- **MeshAttachment** (Type type, const char \*name=nullptr)
- **MeshAttachment** ([MeshNode](#) &node, const char \*name=nullptr)
- void [clear](#) ()  
*clear attachment*
- void [setType](#) (Type type)  
*attachment type*
- Type **getType** () const
- const char \* **getTypeName** () const
- bool **isUnknown** () const
- bool **isLight** () const
- bool **isCamera** () const
- void [setName](#) (const char \*name)  
*attachment name*
- [String](#) **getName** () const
- void [setNode](#) ([MeshNode](#) &node, bool check=true)  
*attachment node*
- const [MeshNode](#) **getNode** () const
- [MeshNode](#) **getNode** ()
- void [setData](#) (const char \*data)  
*attachment data*
- void **setData** (const [String](#) &data)
- [String](#) **getData** () const
- void [clearParameters](#) ()  
*attachment parameters*
- bool **removeParameter** (const char \*type)
- void **copyParameters** (const [MeshAttachment](#) &attachment)
- uint32\_t **findParameter** (const char \*type) const
- bool **hasParameter** (const char \*type) const

- uint32\_t **getNumParameters** () const
- [String](#) **getParameterType** (uint32\_t index) const
- void **addParameter** (const char \*type, bool value)
  - add attachment parameters*
- void **addParameter** (const char \*type, float32\_t value)
- void **addParameter** (const char \*type, const [Vector4f](#) &vector)
- void **addParameter** (const char \*type, const [Color](#) &color)
- void **addParameter** (const char \*type, const char \*name)
- void **addParameter** (const char \*type, const [String](#) &name)
- [Flags](#) **getParameterFlags** (uint32\_t index) const
  - get attachment parameter by index*
- bool **hasParameterFlag** (uint32\_t index, [Flags](#) flags) const
- bool **hasParameterFlags** (uint32\_t index, [Flags](#) flags) const
- bool **getParameterBool** (uint32\_t index, bool value=false) const
- float32\_t **getParameterScalarf32** (uint32\_t index, float32\_t value=0.0f) const
- const [Vector4f](#) & **getParameterVector4f** (uint32\_t index, const [Vector4f](#) &vector=[Vector4f::zero](#)) const
- const [Color](#) & **getParameterColor** (uint32\_t index, const [Color](#) &color=[Color::white](#)) const
- [String](#) **getParameterName** (uint32\_t index, const [String](#) &name=[String::null](#)) const
- [Flags](#) **getParameterFlags** (const char \*type) const
  - get attachment parameter by type*
- bool **hasParameterFlag** (const char \*type, [Flags](#) flags) const
- bool **hasParameterFlags** (const char \*type, [Flags](#) flags) const
- bool **getParameterBool** (const char \*type, bool value=false) const
- float32\_t **getParameterScalarf32** (const char \*type, float32\_t value=0.0f) const
- const [Vector4f](#) & **getParameterVector4f** (const char \*type, const [Vector4f](#) &vector=[Vector4f::zero](#)) const
- const [Color](#) & **getParameterColor** (const char \*type, const [Color](#) &color=[Color::white](#)) const
- [String](#) **getParameterName** (const char \*type, const [String](#) &name=[String::null](#)) const
- void **setTransform** (const [Vector3f](#) &scale)
  - attachment transform*
- void **setTransform** (const [Matrix4x3f](#) &transform)
- const [Matrix4x3f](#) & **getTransform** () const
- int32\_t **compare** (const [MeshAttachment](#) &attachment) const
  - compare attachments*
- size\_t **getMemory** () const
  - memory usage*

#### Static Public Member Functions

- static const char \* **getTypeName** (Type type)

#### Friends

- class **MeshNode**

#### 5.188.1 Detailed Description

[MeshAttachment](#) class

## 5.189 Tellusim::MeshAttribute Class Reference

```
#include <format/TellusimMesh.h>
```

### Public Member Functions

- **MeshAttribute** (const char \*name=nullptr, uint32\_t index=0)
- **MeshAttribute** (Type type, Format format, const char \*name=nullptr, uint32\_t index=0)
- **MeshAttribute** (Type type, Format format, uint32\_t size, const char \*name=nullptr, uint32\_t index=0)
- **MeshAttribute** (Type type, Format format, uint32\_t size, uint32\_t index)
- void **clear** ()  
*clear attributes*
- void **setName** (const char \*name)  
*attribute name*
- **String** **getName** () const
- void **create** (Type type, Format format, uint32\_t size=0)  
*create attribute*
- Type **getType** () const  
*attribute type*
- const char \* **getTypeName** () const
- bool **isUnknown** () const
- bool **isPosition** () const
- bool **isBasis** () const
- bool **isNormal** () const
- bool **isTangent** () const
- bool **isBinormal** () const
- bool **isSpatial** () const
- bool **isNormalized** () const
- bool **isTexCoord** () const
- bool **isWeights** () const
- bool **isJoints** () const
- bool **isColor** () const
- bool **isVertex** () const
- bool **isCrease** () const
- Format **getFormat** () const  
*attribute format*
- const char \* **getFormatName** () const
- uint32\_t **getComponents** () const
- bool **isPacked** () const
- void **setIndex** (uint32\_t index)  
*attribute index*
- uint32\_t **getIndex** () const
- void **setIndices** (**MeshIndices** &indices)  
*attribute indices*
- const **MeshIndices** **getIndices** () const
- **MeshIndices** **getIndices** ()
- void **setGeometry** (**MeshGeometry** &geometry, bool check=true)  
*attribute geometry*
- const **MeshGeometry** **getGeometry** () const
- **MeshGeometry** **getGeometry** ()
- void **setSize** (uint32\_t size, bool discard=true, bool **clear**=false)  
*attribute size*

- uint32\_t **getSize** () const
- uint32\_t **getStride** () const
- size\_t **getBytes** () const
- void **setData** (const void \*src, uint32\_t size=0, uint32\_t stride=0)
  - attribute data*
- void **setData** (const void \*src, const Array< uint32\_t > &indices, uint32\_t stride=0)
- void **getData** (void \*dest, uint32\_t size=0, uint32\_t stride=0) const
- void **getData** (void \*dest, const MeshIndices &indices, uint32\_t stride=0) const
- void **getData** (void \*dest, const Array< uint32\_t > &indices, uint32\_t stride=0) const
- const void \* **getData** () const
- void \* **getData** ()
- template<class Type >
  - void **set** (const Type &value)
    - attribute values*
- template<class Type >
  - void **set** (uint32\_t index, const Type &value)
- template<class Type >
  - const Type & **get** (uint32\_t index) const
- template<class Type >
  - Type & **get** (uint32\_t index)
- void **setValue** (uint32\_t index, const void \*src, size\_t size)
  - attribute value*
- void **getValue** (uint32\_t index, void \*dest, size\_t size) const
- const void \* **getPtr** (uint32\_t index) const
  - attribute pointers*
- void \* **getPtr** (uint32\_t index)
- int32\_t **compare** (const MeshAttribute &attribute, const Matrix4x3f &transform=Matrix4x3f::identity, float32\_t threshold=1e-6f, bool spatial=true) const
  - compare attributes*
- void **addAttribute** (const MeshAttribute &attribute)
  - add attribute*
- bool **setTransform** (const Matrix4x3f &transform)
  - apply transform*
- bool **morphAttribute** (const MeshAttribute &attribute, float32\_t k)
  - morph attribute*
- bool **packAttributes** (const MeshAttribute &attribute\_0, const MeshAttribute &attribute\_1, Format format)
  - pack attributes*
- bool **unpackAttributes** (MeshAttribute &attribute\_0, MeshAttribute &attribute\_1) const
- MeshAttribute **optimizeAttribute** (MeshIndices &indices) const
  - optimize attribute by removing duplicates*
- MeshAttribute **toDirect** (const MeshIndices &indices) const
  - convert attribute to direct*
- MeshAttribute **toFormat** (Format format) const
  - convert attribute to format*
- MeshAttribute **toType** (Type type) const
  - convert attribute to type*
- Matrix4x3f **getCovarianceMatrix** () const
  - covariance matrix*
- Matrix4x3f **getMinTransform** () const
  - minimal bound transform*
- BoundingBoxf **getBoundingBox** () const
  - attribute bound box*

- [BoundSpheref](#) [getBoundSphere](#) () const  
*attribute bound sphere*
- [size\\_t](#) [getMemory](#) () const  
*memory usage*

#### Static Public Member Functions

- static const char \* [getTypeName](#) (Type type)

#### Friends

- class **MeshGeometry**

#### 5.189.1 Detailed Description

[MeshAttribute](#) class

### 5.190 Tellusim::MeshGeometry Class Reference

```
#include <format/TellusimMesh.h>
```

#### Public Member Functions

- **MeshGeometry** (const char \*name=nullptr)
- **MeshGeometry** ([Mesh](#) &mesh, const char \*name=nullptr)
- void [clear](#) ()  
*clear geometry*
- void [setName](#) (const char \*name)  
*geometry name*
- [String](#) [getName](#) () const
- [uint32\\_t](#) [getIndex](#) () const  
*geometry index*
- void [setMesh](#) ([Mesh](#) &mesh, bool check=true)  
*geometry mesh*
- const [Mesh](#) [getMesh](#) () const
- [Mesh](#) [getMesh](#) ()
- [uint32\\_t](#) [setParent0](#) ([MeshGeometry](#) &parent, bool check=true)  
*geometry parent*
- [uint32\\_t](#) [setParent1](#) ([MeshGeometry](#) &parent, bool check=true)
- const [MeshGeometry](#) [getParent0](#) () const
- const [MeshGeometry](#) [getParent1](#) () const
- [MeshGeometry](#) [getParent0](#) ()
- [MeshGeometry](#) [getParent1](#) ()
- bool [isRoot](#) () const
- void [reserveChildren](#) ([uint32\\_t](#) num\_children)  
*geometry children*
- [uint32\\_t](#) [addChild0](#) ([MeshGeometry](#) &child, bool check=true)
- [uint32\\_t](#) [addChild1](#) ([MeshGeometry](#) &child, bool check=true)



- bool **removeChild** ([MeshGeometry](#) &child)
- void **releaseChildren** ()
- uint32\_t **findChild** (const [MeshGeometry](#) &child) const
- uint32\_t **getNumChildren** () const
- const Array< [MeshGeometry](#) > **getChildren** () const
- Array< [MeshGeometry](#) > **getChildren** ()
- const [MeshGeometry](#) **getChild** (uint32\_t index) const
- [MeshGeometry](#) **getChild** (uint32\_t index)
- void **clearIndices** ()
- geometry indices*
- void **reserveIndices** (uint32\_t num\_indices)
- uint32\_t **addIndices** ([MeshIndices](#) &indices, bool check=true)
- bool **removeIndices** ([MeshIndices](#) &indices)
- bool **replaceIndices** ([MeshIndices](#) &old\_indices, [MeshIndices](#) &indices)
- uint32\_t **findIndices** (const [MeshIndices](#) &indices) const
- uint32\_t **findIndices** ([MeshIndices::Type](#) type) const
- bool **hasIndices** ([MeshIndices::Type](#) type) const
- bool **hasSolidIndices** () const
- uint32\_t **getNumIndices** ([MeshIndices::Type](#) type) const
- const [MeshIndices](#) **getIndices** ([MeshIndices::Type](#) type) const
- [MeshIndices](#) **getIndices** ([MeshIndices::Type](#) type)
- uint32\_t **getNumIndices** () const
- const Array< [MeshIndices](#) > **getIndices** () const
- Array< [MeshIndices](#) > **getIndices** ()
- const [MeshIndices](#) **getIndices** (uint32\_t index) const
- [MeshIndices](#) **getIndices** (uint32\_t index)
- void **clearAttributes** ()
- geometry attributes*
- void **reserveAttributes** (uint32\_t num\_attributes)
- uint32\_t **addAttribute** ([MeshAttribute](#) &attribute, bool check=true)
- uint32\_t **addAttribute** ([MeshAttribute](#) &attribute, [MeshIndices](#) &indices, bool check=true)
- bool **removeAttribute** ([MeshAttribute](#) &attribute)
- bool **replaceAttribute** ([MeshAttribute](#) &old\_attribute, [MeshAttribute](#) &attribute)
- bool **replaceAttributeIndices** (const [MeshIndices](#) &old\_indices, [MeshIndices](#) &indices)
- uint32\_t **findAttribute** ([MeshAttribute::Type](#) type, Format format, uint32\_t index=0) const
- uint32\_t **findAttribute** ([MeshAttribute::Type](#) type, uint32\_t index=0) const
- uint32\_t **findAttribute** (const char \*name, uint32\_t index=0) const
- uint32\_t **findAttribute** (const [MeshAttribute](#) &attribute) const
- bool **hasAttribute** ([MeshAttribute::Type](#) type, Format format, uint32\_t index=0) const
- bool **hasAttribute** ([MeshAttribute::Type](#) type, uint32\_t index=0) const
- bool **hasAttribute** (const char \*name, uint32\_t index=0) const
- bool **hasAttribute** (const [MeshAttribute](#) &attribute) const
- uint32\_t **getNumAttributes** ([MeshAttribute::Type](#) type) const
- uint32\_t **getNumAttributes** (const [MeshIndices](#) &indices) const
- const [MeshAttribute](#) **getAttribute** ([MeshAttribute::Type](#) type, uint32\_t index=0) const
- [MeshAttribute](#) **getAttribute** ([MeshAttribute::Type](#) type, uint32\_t index=0)
- uint32\_t **getNumAttributes** () const
- const Array< [MeshAttribute](#) > **getAttributes** () const
- Array< [MeshAttribute](#) > **getAttributes** ()
- const [MeshAttribute](#) **getAttribute** (uint32\_t index) const
- [MeshAttribute](#) **getAttribute** (uint32\_t index)
- void **clearJoints** ()
- geometry joints*
- void **reserveJoints** (uint32\_t num\_joints)

- uint32\_t **addJoint** ([MeshJoint](#) &joint, bool check=true)
- uint32\_t **addJoint** ([MeshJoint](#) &joint, [MeshNode](#) &node, bool check=true)
- bool **removeJoint** ([MeshJoint](#) &joint)
- bool **replaceJoint** ([MeshJoint](#) &old\_joint, [MeshJoint](#) &joint)
- uint32\_t **findJoint** (const [MeshJoint](#) &joint) const
- uint32\_t **findJoint** (const [MeshNode](#) &node) const
- uint32\_t **findJoint** (const char \*name) const
- uint32\_t **getNumJoints** () const
- const Array< [MeshJoint](#) > **getJoints** () const
- Array< [MeshJoint](#) > **getJoints** ()
- const [MeshJoint](#) **getJoint** (uint32\_t index) const
- [MeshJoint](#) **getJoint** (uint32\_t index)
- void **clearMaterials** ()
  - geometry materials*
  - void **reserveMaterials** (uint32\_t num\_materials)
  - uint32\_t **addMaterial** ([MeshMaterial](#) &material, bool check=true)
  - uint32\_t **addMaterial** ([MeshMaterial](#) &material, [MeshIndices](#) &indices, bool check=true)
  - bool **removeMaterial** ([MeshMaterial](#) &material)
  - bool **replaceMaterial** ([MeshMaterial](#) &old\_material, [MeshMaterial](#) &material)
  - uint32\_t **findMaterial** (const [MeshMaterial](#) &material) const
  - uint32\_t **findMaterial** (const char \*name) const
  - uint32\_t **getNumMaterials** () const
  - const Array< [MeshMaterial](#) > **getMaterials** () const
  - Array< [MeshMaterial](#) > **getMaterials** ()
  - const [MeshMaterial](#) **getMaterial** (uint32\_t index) const
  - [MeshMaterial](#) **getMaterial** (uint32\_t index)
  - void **setBoundingBox** (const [BoundingBox](#) &box)
    - geometry bound box*
    - const [BoundingBox](#) & **getBoundingBox** () const
    - void **setBoundSphere** (const [BoundSpheref](#) &sphere)
      - geometry bound sphere*
      - const [BoundSpheref](#) & **getBoundSphere** () const
      - bool **setTransform** (const [Vector3f](#) &scale)
        - geometry transform*
        - bool **setTransform** (const [Matrix4x3f](#) &transform, bool apply=false)
        - const [Matrix4x3f](#) & **getTransform** () const
        - void **setJointTransform** (const [Matrix4x3f](#) &itransform)
          - geometry inverse joint transform*
          - const [Matrix4x3f](#) & **getJointTransform** () const
          - void **setMinVisibility** (float32\_t distance)
            - visibility range*
            - void **setMaxVisibility** (float32\_t distance)
            - void **setVisibilityRange** (float32\_t min, float32\_t max)
            - float32\_t **getMinVisibility** () const
            - float32\_t **getMaxVisibility** () const
            - bool **hasVisibilityRange** () const
            - void **setVisibilityError** (float32\_t error)
              - visibility error*
              - float32\_t **getVisibilityError** () const
              - bool **createBounds** (bool force=false, uint32\_t position=Maxu32)
              - uint32\_t **createBasis** (bool force=false, uint32\_t position=Maxu32, uint32\_t normal=Maxu32, uint32\_t tangent=Maxu32, bool append=false)
              - uint32\_t **createBasis** (float32\_t angle, bool force=false, uint32\_t position=Maxu32, uint32\_t normal=Maxu32, uint32\_t tangent=Maxu32, bool append=false)

- uint32\_t [createNormals](#) (bool force=false, uint32\_t position=Maxu32, bool append=false)
- uint32\_t **createNormals** (float32\_t angle, bool force=false, uint32\_t position=Maxu32, bool append=false)
- uint32\_t [createTangents](#) (bool force=false, uint32\_t position=Maxu32, uint32\_t normal=Maxu32, uint32\_t texcoord=Maxu32, bool append=false)
- uint32\_t [createIslands](#) (uint32\_t max\_attributes, uint32\_t max\_primitives, bool force=false, uint32\_t index=Maxu32, uint32\_t position=Maxu32, bool append=false)
- bool [optimizeIndices](#) (uint32\_t cache=32, bool transparent=false, uint32\_t index=Maxu32, uint32\_t position=Maxu32)
- bool [optimizeAttributes](#) (uint32\_t material=Maxu32)
- void [optimizeMaterials](#) ()  
*optimize materials remove duplicates*
- bool [packAttributes](#) (bool remove=true)  
*pack attributes (morph targets, texture coordinates and vertex colors)*
- bool **unpackAttributes** (bool remove=true)
- int32\_t [compare](#) (const [MeshGeometry](#) &geometry, const [Matrix4x3f](#) &transform=Matrix4x3f::identity, float32\_t threshold=1e-6f, bool spatial=true) const  
*compare geometries*
- bool [isOptimized](#) () const  
*optimized geometry flag (if geometry contains single indices).*
- bool [validate](#) () const  
*validate geometry*
- size\_t [getMemory](#) () const  
*memory usage*

## Friends

- class **Mesh**

### 5.190.1 Detailed Description

[MeshGeometry](#) class

### 5.190.2 Member Function Documentation

#### 5.190.2.1 createBounds()

```
bool Tellusim::MeshGeometry::createBounds (
    bool force = false,
    uint32_t position = Maxu32 )
```

create bounds

#### Parameters

|                 |                           |
|-----------------|---------------------------|
| <i>force</i>    | Force bounding creation.  |
| <i>position</i> | Position attribute index. |

### 5.190.2.2 createBasis()

```
uint32_t Tellusim::MeshGeometry::createBasis (
    bool force = false,
    uint32_t position = Maxu32,
    uint32_t normal = Maxu32,
    uint32_t tangent = Maxu32,
    bool append = false )
```

create tangent basis

#### Parameters

|                 |                             |
|-----------------|-----------------------------|
| <i>force</i>    | Force basis creation.       |
| <i>position</i> | Position attribute index.   |
| <i>normal</i>   | Normal attribute index.     |
| <i>tangent</i>  | Tangent attribute index.    |
| <i>append</i>   | Append new basis attribute. |

#### Returns

[Basis](#) attribute index.

### 5.190.2.3 createNormals()

```
uint32_t Tellusim::MeshGeometry::createNormals (
    bool force = false,
    uint32_t position = Maxu32,
    bool append = false )
```

create normals

#### Parameters

|                 |                              |
|-----------------|------------------------------|
| <i>force</i>    | Force normals creation.      |
| <i>position</i> | Position attribute index.    |
| <i>angle</i>    | Smoothing angle in degrees.  |
| <i>append</i>   | Append new normal attribute. |

#### Returns

Normal attribute index.

#### 5.190.2.4 createTangents()

```
uint32_t Tellusim::MeshGeometry::createTangents (
    bool force = false,
    uint32_t position = Maxu32,
    uint32_t normal = Maxu32,
    uint32_t texcoord = Maxu32,
    bool append = false )
```

create tangents

##### Parameters

|                 |                               |
|-----------------|-------------------------------|
| <i>force</i>    | Force tangents creation.      |
| <i>position</i> | Position attribute index.     |
| <i>normal</i>   | Normal attribute index.       |
| <i>texcoord</i> | TexCoord attribute index.     |
| <i>append</i>   | Append new tangent attribute. |

##### Returns

Tangent attribute index.

#### 5.190.2.5 createlands()

```
uint32_t Tellusim::MeshGeometry::createIslands (
    uint32_t max_attributes,
    uint32_t max_primitives,
    bool force = false,
    uint32_t index = Maxu32,
    uint32_t position = Maxu32,
    bool append = false )
```

create islands

##### Parameters

|                       |                                          |
|-----------------------|------------------------------------------|
| <i>max_attributes</i> | Maximum number of attributes per island. |
| <i>max_primitives</i> | Maximum number of primitives per island. |
| <i>index</i>          | Indices index to update.                 |
| <i>position</i>       | Position attribute index.                |
| <i>append</i>         | Append new island indices.               |

##### Returns

Island indices index.

### 5.190.2.6 optimizeIndices()

```
bool Tellusim::MeshGeometry::optimizeIndices (
    uint32_t cache = 32,
    bool transparent = false,
    uint32_t index = Maxu32,
    uint32_t position = Maxu32 )
```

optimize indices

#### Parameters

|                    |                            |
|--------------------|----------------------------|
| <i>cache</i>       | Vertex cache size.         |
| <i>transparent</i> | Optimize for transparency. |
| <i>index</i>       | Indices index to optimize. |
| <i>position</i>    | Position attribute index.  |

### 5.190.2.7 optimizeAttributes()

```
bool Tellusim::MeshGeometry::optimizeAttributes (
    uint32_t material = Maxu32 )
```

optimize attributes and make single indices.

#### Parameters

|                 |                 |
|-----------------|-----------------|
| <i>material</i> | material index. |
|-----------------|-----------------|

## 5.191 Tellusim::MeshIndices Class Reference

```
#include <format/TellusimMesh.h>
```

#### Public Member Functions

- **MeshIndices** (const char \*name=nullptr)
- **MeshIndices** (Type type, Format format, const char \*name=nullptr)
- **MeshIndices** (Type type, Format format, uint32\_t size, const char \*name=nullptr)
- void **clear** ()
  - clear indices*
- void **setName** (const char \*name)
  - indices name*
- **String getName** () const
- void **create** (Type type, Format format, uint32\_t size=0)
  - create indices*
- Type **getType** () const
  - indices type*
- const char \* **getTypeName** () const

- bool **isUnknown** () const
- bool **isPoint** () const
- bool **isLine** () const
- bool **isTriangle** () const
- bool **isQuadrilateral** () const
- bool **isTetrahedron** () const
- bool **isPrimitive** () const
- bool **isSolid** () const
- bool **isVolume** () const
- bool **isMaterial** () const
- bool **isGroup** () const
- bool **isJoint** () const
- bool **isEdge** () const
- uint32\_t **getPrimitiveSize** () const
- Format **getFormat** () const
- indices format*
- const char \* **getFormatName** () const
- void **setGeometry** (MeshGeometry &geometry, bool check=true)
- indices geometry*
- const MeshGeometry **getGeometry** () const
- MeshGeometry **getGeometry** ()
- void **setSize** (uint32\_t size, bool discard=true, bool clear=false)
- indices size*
- uint32\_t **getSize** () const
- uint32\_t **getStride** () const
- size\_t **getBytes** () const
- void **setData** (uint32\_t value, uint32\_t size=0, uint32\_t offset=0)
- indices data*
- void **setData** (const void \*src, Format format=FormatUnknown, uint32\_t size=0, uint32\_t repeat=1)
- void **getData** (void \*dest, Format format=FormatUnknown, uint32\_t size=0, uint32\_t repeat=1) const
- const void \* **getData** () const
- void \* **getData** ()
- void **set** (uint32\_t index, uint32\_t value)
- indices values*
- void **set** (uint32\_t index, uint32\_t value\_0, uint32\_t value\_1)
- void **set** (uint32\_t index, uint32\_t value\_0, uint32\_t value\_1, uint32\_t value\_2)
- void **set** (uint32\_t index, uint32\_t value\_0, uint32\_t value\_1, uint32\_t value\_2, uint32\_t value\_3)
- uint32\_t **get** (uint32\_t index) const
- void **get** (uint32\_t index, uint32\_t &value\_0, uint32\_t &value\_1) const
- void **get** (uint32\_t index, uint32\_t &value\_0, uint32\_t &value\_1, uint32\_t &value\_2) const
- void **get** (uint32\_t index, uint32\_t &value\_0, uint32\_t &value\_1, uint32\_t &value\_2, uint32\_t &value\_3) const
- const void \* **getPtr** (uint32\_t index) const
- indices pointers*
- void \* **getPtr** (uint32\_t index)
- bool **isDirect** () const
- direct indices flag*
- bool **isUniform** () const
- uniform indices flag*
- uint32\_t **getMinIndex** () const
- indices range*
- uint32\_t **getMaxIndex** () const
- int32\_t **compare** (const MeshIndices &indices) const
- compare indices*

- void **addIndices** (const [MeshIndices](#) &indices, uint32\_t offset, bool expand=false)  
*add indices*
- [MeshIndices toFormat](#) (Format format) const  
*convert indices to format*
- [MeshIndices toType](#) (Type type) const  
*convert indices to type*
- [MeshIndices toType](#) (Type type, const [MeshAttribute](#) &position\_attribute) const
- size\_t **getMemory** () const  
*memory usage*

#### Static Public Member Functions

- static const char \* **getTypeName** (Type type)

#### Friends

- class **MeshGeometry**

#### 5.191.1 Detailed Description

[MeshIndices](#) class

### 5.192 Tellusim::MeshJoint Class Reference

```
#include <format/TellusimMesh.h>
```

#### Public Member Functions

- **MeshJoint** (const char \*name=nullptr)
- **MeshJoint** ([MeshGeometry](#) &geometry, const char \*name=nullptr)
- void **clear** ()  
*clear joint*
- void **setName** (const char \*name)  
*joint name*
- [String](#) **getName** () const
- void **setNode** ([MeshNode](#) &node)  
*joint node*
- const [MeshNode](#) **getNode** () const
- [MeshNode](#) **getNode** ()
- uint32\_t **getNodeIndex** () const
- const [Matrix4x3d](#) & **getLocalTransform** () const
- const [Matrix4x3d](#) & **getGlobalTransform** () const
- void **setIndices** ([MeshIndices](#) &indices)  
*joint indices*
- const [MeshIndices](#) **getIndices** () const
- [MeshIndices](#) **getIndices** ()
- void **setGeometry** ([MeshGeometry](#) &geometry, bool check=true)  
*joint geometry*



- const [MeshGeometry](#) **getGeometry** () const
- [MeshGeometry](#) **getGeometry** ()
- void **setBoundingBox** (const [BoundingBoxf](#) &box)  
*joint bound box*
- const [BoundingBoxf](#) & **getBoundingBox** () const
- void **setBoundSphere** (const [BoundSphref](#) &sphere)  
*joint bound sphere*
- const [BoundSphref](#) & **getBoundSphere** ()
- void **setTransform** (const [Matrix4x3f](#) &itransform)  
*inverse joint transform*
- const [Matrix4x3f](#) & **getTransform** () const
- int32\_t **compare** (const [MeshJoint](#) &joint) const  
*compare joints*
- size\_t **getMemory** () const  
*memory usage*

#### Friends

- class **MeshGeometry**

#### 5.192.1 Detailed Description

[MeshJoint](#) class

### 5.193 Tellusim::MeshModel::Meshlet Struct Reference

#### Public Attributes

- uint32\_t **num\_primitives**
- uint32\_t **num\_vertices**
- uint32\_t **base\_index**
- uint32\_t **base\_vertex**
- float32\_t **bound\_sphere** [4]
- float32\_t **normal\_angle** [4]

### 5.194 Tellusim::MeshMaterial Class Reference

```
#include <format/TellusimMesh.h>
```

#### Public Types

- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagBool** = (1 << 0),  
**FlagScalarf32** = (1 << 1),  
**FlagVector4f** = (1 << 2),  
**FlagMatrix3x2f** = (1 << 3),  
**FlagColor** = (1 << 4),  
**FlagName** = (1 << 5),  
**FlagLayout** = (1 << 6),  
**FlagBlob** = (1 << 7),  
**FlagImage** = (1 << 8),  
**FlagTexture** = (FlagName | FlagBlob | FlagImage) }  
*Material flags.*

## Public Member Functions

- **MeshMaterial** (const char \*name=nullptr)
- **MeshMaterial** ([MeshGeometry](#) &geometry, const char \*name=nullptr)
- void **clear** ()  
*clear material*
- void **setName** (const char \*name)  
*material name*
- [String](#) **getName** () const
- uint32\_t **getIndex** () const  
*material index*
- void **setIndices** ([MeshIndices](#) &indices)  
*material indices*
- const [MeshIndices](#) **getIndices** () const
- [MeshIndices](#) **getIndices** ()
- void **setGeometry** ([MeshGeometry](#) &geometry, bool check=true)  
*material geometry*
- const [MeshGeometry](#) **getGeometry** () const
- [MeshGeometry](#) **getGeometry** ()
- void **setData** (const char \*data)  
*material data*
- void **setData** (const [String](#) &data)
- [String](#) **getData** () const
- void **clearParameters** ()  
*material parameters*
- bool **removeParameter** (const char \*type)
- void **copyParameters** (const [MeshMaterial](#) &material)
- uint32\_t **findParameter** (const char \*type) const
- bool **hasParameter** (const char \*type) const
- uint32\_t **getNumParameters** () const
- [String](#) **getParameterType** (uint32\_t index) const
- void **addParameter** (const char \*type, bool value)  
*add material parameters*
- void **addParameter** (const char \*type, float32\_t value)
- void **addParameter** (const char \*type, const [Vector4f](#) &value)
- void **addParameter** (const char \*type, const [Matrix3x2f](#) &value)
- void **addParameter** (const char \*type, const [Color](#) &color)
- void **addParameter** (const char \*type, const char \*name, const char \*layout=nullptr)
- void **addParameter** (const char \*type, const [String](#) &name, const char \*layout=nullptr)
- void **addParameter** (const char \*type, const [Image](#) &image, const char \*layout=nullptr)
- void **addParameter** (const char \*type, [Blob](#) &blob, const char \*layout=nullptr)
- [Flags](#) **getParameterFlags** (uint32\_t index) const  
*get material parameter by index*
- bool **hasParameterFlag** (uint32\_t index, [Flags](#) flags) const
- bool **hasParameterFlags** (uint32\_t index, [Flags](#) flags) const
- bool **getParameterBool** (uint32\_t index, bool value=false) const
- float32\_t **getParameterScalarf32** (uint32\_t index, float32\_t value=0.0f) const
- const [Vector4f](#) & **getParameterVector4f** (uint32\_t index, const [Vector4f](#) &vector=[Vector4f::zero](#)) const
- const [Matrix3x2f](#) & **getParameterMatrix3x2f** (uint32\_t index, const [Matrix3x2f](#) &matrix=[Matrix3x2f::identity](#)) const
- const [Color](#) & **getParameterColor** (uint32\_t index, const [Color](#) &color=[Color::white](#)) const
- [String](#) **getParameterName** (uint32\_t index, const [String](#) &name=[String::null](#)) const
- [String](#) **getParameterLayout** (uint32\_t index, const [String](#) &layout=[String::null](#)) const

- const [Image](#) **getParameterImage** (uint32\_t index) const
- [Blob](#) **getParameterBlob** (uint32\_t index) const
- [Image](#) **getParameterImage** (uint32\_t index)
- [Blob](#) **getParameterBlob** (uint32\_t index)
- [Flags](#) **getParameterFlags** (const char \*type) const  
*get material parameter by type*
- bool **hasParameterFlag** (const char \*type, [Flags](#) flags) const
- bool **hasParameterFlags** (const char \*type, [Flags](#) flags) const
- bool **getParameterBool** (const char \*type, bool value=false) const
- float32\_t **getParameterScalarf32** (const char \*type, float32\_t value=0.0f) const
- const [Vector4f](#) & **getParameterVector4f** (const char \*type, const [Vector4f](#) &vector=[Vector4f::zero](#)) const
- const [Matrix3x2f](#) & **getParameterMatrix3x2f** (const char \*type, const [Matrix3x2f](#) &matrix=[Matrix3x2f](#)::identity) const
- const [Color](#) & **getParameterColor** (const char \*type, const [Color](#) &color=[Color::white](#)) const
- [String](#) **getParameterName** (const char \*type, const [String](#) &name=[String::null](#)) const
- [String](#) **getParameterLayout** (const char \*type, const [String](#) &layout=[String::null](#)) const
- int32\_t **compare** (const [MeshMaterial](#) &material) const  
*compare materials*
- size\_t **getMemory** () const  
*memory usage*

#### Friends

- class **MeshGeometry**

#### 5.194.1 Detailed Description

[MeshMaterial](#) class

#### 5.195 Tellusim::MeshModel Class Reference

```
#include <graphics/TellusimMeshModel.h>
```

#### Classes

- struct [Meshlet](#)

## Public Types

- enum **Flags** {  
**FlagNone** = 0,  
**FlagDirect** = (1 << 0),  
**FlagVerbose** = (1 << 1),  
**FlagOptimize** = (1 << 2),  
**FlagMaterials** = (1 << 3),  
**FlagIndices10** = (1 << 4),  
**FlagIndices16** = (1 << 5),  
**FlagIndices32** = (1 << 6),  
**FlagMeshlet64x84** = (1 << 7),  
**FlagMeshlet64x126** = (1 << 8),  
**FlagMeshlet96x169** = (1 << 9),  
**FlagMeshlet128x212** = (1 << 10),  
**FlagBufferWrite** = (1 << 11),  
**FlagBufferSource** = (1 << 12),  
**FlagBufferStorage** = (1 << 13),  
**FlagBufferTracing** = (1 << 14),  
**FlagBufferAddress** = (1 << 15),  
**FlagBufferTexel** = (1 << 16),  
**FlagMeshlets** = (FlagMeshlet64x84 | FlagMeshlet64x126 | FlagMeshlet96x169 | FlagMeshlet128x212),  
**DefaultFlags** = (FlagVerbose | FlagMaterials),  
**NumFlags** = 17 }  
*Model flags.*
- using **CreateCallback** = Function< bool(const void \*src, size\_t size, bool owner)>  
*create buffer callbacks*

## Public Member Functions

- void **clear** ()  
*clear model*
- bool **isCreated** () const  
*check model*
- **Flags getFlags** () const  
*model flags*
- bool **hasFlag** (**Flags** flags) const
- bool **hasFlags** (**Flags** flags) const
- bool **load** (const **Device** &device, const **Pipeline** &pipeline, const char \*name, **Flags** flags=DefaultFlags, **Async** \*async=nullptr)  
*load model*
- bool **load** (const **Device** &device, const **Pipeline** &pipeline, **Stream** &stream, **Flags** flags=DefaultFlags, **Async** \*async=nullptr)
- bool **create** (const **Device** &device, const **Pipeline** &pipeline, const char \*name, **Flags** flags=DefaultFlags)  
*create model*
- bool **create** (const **Device** &device, const **Pipeline** &pipeline, const **Mesh** &mesh, **Flags** flags=DefaultFlags)
- bool **create** (const **Device** &device, const **Pipeline** &pipeline, const **MeshGeometry** &geometry, **Flags** flags=DefaultFlags)
- bool **create** (const **Device** &device, const **Pipeline** &pipeline, const Array< **MeshGeometry** > &geometries, **Flags** flags=DefaultFlags)
- void **setVertexBufferCallback** (const **CreateCallback** &func)
- void **setIndexBufferCallback** (const **CreateCallback** &func)
- void **setMeshBufferCallback** (const **CreateCallback** &func)
- void **setBuffers** (**Command** &command, uint32\_t index=0, const **Pipeline** \*pipeline=nullptr) const

- set model buffers*
- void **draw** ([Command](#) &command) const
- draw model*
- void **draw** ([Command](#) &command, uint32\_t geometry) const
- void **draw** ([Command](#) &command, uint32\_t geometry, uint32\_t material) const
- void **drawInstanced** ([Command](#) &command, uint32\_t geometry, uint32\_t num\_instances, uint32\_t base\_instance=0) const
- draw instanced model*
- void **drawInstanced** ([Command](#) &command, uint32\_t geometry, uint32\_t material, uint32\_t num\_instances, uint32\_t base\_instance) const
- uint32\_t **getNumVertices** () const
- vertices buffer*
- uint32\_t **getNumVertexBuffers** () const
- uint32\_t **getVertexBufferStride** (uint32\_t index) const
- size\_t **getVertexBufferOffset** (uint32\_t index) const
- [Buffer](#) **getVertexBuffer** () const
- uint32\_t **getNumIndices** () const
- indices buffer*
- Format **getIndexFormat** () const
- [Buffer](#) **getIndexBuffer** () const
- uint32\_t **getNumMeshlets** () const
- meshlets buffer*
- [Buffer](#) **getMeshletBuffer** () const
- uint32\_t **getNumGeometries** () const
- geometries*
- uint32\_t **getNumGeometryIndices** (uint32\_t geometry) const
- uint32\_t **getNumGeometryVertices** (uint32\_t geometry) const
- uint32\_t **getNumGeometryMeshlets** (uint32\_t geometry) const
- uint32\_t **getGeometryBaseIndex** (uint32\_t geometry) const
- uint32\_t **getGeometryBaseVertex** (uint32\_t geometry) const
- uint32\_t **getGeometryBaseMeshlet** (uint32\_t geometry) const
- uint32\_t **getNumMaterials** (uint32\_t geometry) const
- geometry materials*
- uint32\_t **getNumMaterialIndices** (uint32\_t geometry, uint32\_t material) const
- uint32\_t **getNumMaterialVertices** (uint32\_t geometry, uint32\_t material) const
- uint32\_t **getNumMaterialMeshlets** (uint32\_t geometry, uint32\_t material) const
- uint32\_t **getMaterialBaseIndex** (uint32\_t geometry, uint32\_t material) const
- uint32\_t **getMaterialBaseVertex** (uint32\_t geometry, uint32\_t material) const
- uint32\_t **getMaterialBaseMeshlet** (uint32\_t geometry, uint32\_t material) const
- size\_t **getMemory** () const
- memory usage*

#### Protected Member Functions

- virtual bool **create\_vertex\_buffer** (const [Device](#) &device, const void \*src, size\_t size, bool owner)
- virtual bool **create\_index\_buffer** (const [Device](#) &device, const void \*src, size\_t size, bool owner)
- virtual bool **create\_meshlet\_buffer** (const [Device](#) &device, const void \*src, size\_t size, bool owner)

#### 5.195.1 Detailed Description

[MeshModel](#) class

## 5.195.2 Member Function Documentation

### 5.195.2.1 `create_vertex_buffer()`

```
virtual bool Tellusim::MeshModel::create_vertex_buffer (  
    const Device & device,  
    const void * src,  
    size_t size,  
    bool owner ) [protected], [virtual]
```

create vertex buffer

#### Parameters

|              |                                                                                  |
|--------------|----------------------------------------------------------------------------------|
| <i>src</i>   | Vertex data.                                                                     |
| <i>size</i>  | Vertex data size.                                                                |
| <i>owner</i> | If true, the vertex data must be freed with the <a href="#">Allocator</a> class. |

### 5.195.2.2 `create_index_buffer()`

```
virtual bool Tellusim::MeshModel::create_index_buffer (  
    const Device & device,  
    const void * src,  
    size_t size,  
    bool owner ) [protected], [virtual]
```

create index buffer

#### Parameters

|              |                                                                                 |
|--------------|---------------------------------------------------------------------------------|
| <i>src</i>   | Index data.                                                                     |
| <i>size</i>  | Index data size.                                                                |
| <i>owner</i> | If true, the index data must be freed with the <a href="#">Allocator</a> class. |

### 5.195.2.3 `create_meshlet_buffer()`

```
virtual bool Tellusim::MeshModel::create_meshlet_buffer (  
    const Device & device,  
    const void * src,  
    size_t size,  
    bool owner ) [protected], [virtual]
```

create meshlet buffer

#### Parameters

|              |                                                                                   |
|--------------|-----------------------------------------------------------------------------------|
| <i>src</i>   | <a href="#">Meshlet</a> data.                                                     |
| <i>size</i>  | <a href="#">Meshlet</a> data size.                                                |
| <i>owner</i> | If true, the meshlet data must be freed with the <a href="#">Allocator</a> class. |

## 5.196 Tellusim::MeshNode Class Reference

```
#include <format/TellusimMesh.h>
```

## Public Member Functions

- **MeshNode** (const char \*name=nullptr)
- **MeshNode** ([Mesh](#) &mesh, const char \*name=nullptr)
- **MeshNode** ([MeshNode](#) \*parent, const char \*name=nullptr)
- **MeshNode** ([Mesh](#) &mesh, [MeshNode](#) \*parent, const char \*name=nullptr)
- void **clear** ()  
*clear node*
- [MeshNode](#) **clone** ([Mesh](#) &mesh)  
*clone node*
- void **setName** (const char \*name)  
*node name*
- [String](#) **getName** () const
- uint32\_t **getIndex** () const  
*node index*
- void **setMesh** ([Mesh](#) &mesh, bool check=true)  
*node mesh*
- const [Mesh](#) **getMesh** () const
- [Mesh](#) **getMesh** ()
- uint32\_t **setParent** ([MeshNode](#) &parent, bool check=true)  
*node parent*
- const [MeshNode](#) **getParent** () const
- [MeshNode](#) **getParent** ()
- bool **isRoot** () const
- void **reserveChildren** (uint32\_t num\_children)  
*node children*
- uint32\_t **addChild** ([MeshNode](#) &child, bool check=true)
- bool **removeChild** ([MeshNode](#) &child)
- void **releaseChildren** ()
- uint32\_t **findChild** (const [MeshNode](#) &child) const
- uint32\_t **getNumChildren** () const
- const Array< [MeshNode](#) > **getChildren** () const
- Array< [MeshNode](#) > **getChildren** ()
- const [MeshNode](#) **getChild** (uint32\_t index) const
- [MeshNode](#) **getChild** (uint32\_t index)
- void **clearGeometries** ()  
*node geometries*
- void **reserveGeometries** (uint32\_t num\_geometries)
- uint32\_t **addGeometry** ([MeshGeometry](#) &geometry, bool check=true)
- bool **removeGeometry** ([MeshGeometry](#) &geometry)
- bool **replaceGeometry** ([MeshGeometry](#) &old\_geometry, [MeshGeometry](#) &geometry)
- uint32\_t **findGeometry** (const [MeshGeometry](#) &geometry) const
- uint32\_t **getNumGeometries** () const
- const Array< [MeshGeometry](#) > **getGeometries** () const
- Array< [MeshGeometry](#) > **getGeometries** ()
- const [MeshGeometry](#) **getGeometry** (uint32\_t index) const
- [MeshGeometry](#) **getGeometry** (uint32\_t index)
- void **clearAttachments** ()

*node attachments*

- void **reserveAttachments** (uint32\_t num\_attachments)
- uint32\_t **addAttachment** ([MeshAttachment](#) &attachment, bool check=true)
- bool **removeAttachment** ([MeshAttachment](#) &attachment)
- bool **replaceAttachment** ([MeshAttachment](#) &old\_attachment, [MeshAttachment](#) &attachment)
- uint32\_t **findAttachment** (const [MeshAttachment](#) &attachment) const
- uint32\_t **findAttachment** (const char \*name) const
- uint32\_t **getNumAttachments** () const
- const Array< [MeshAttachment](#) > **getAttachments** () const
- Array< [MeshAttachment](#) > **getAttachments** ()
- const [MeshAttachment](#) **getAttachment** (uint32\_t index) const
- [MeshAttachment](#) **getAttachment** (uint32\_t index)
- void **setLocalTransform** (const [Matrix4x3d](#) &transform)

*local transform*

- const [Matrix4x3d](#) & **getLocalTransform** () const
- void **setGlobalTransform** (const [Matrix4x3d](#) &transform)

*global transform*

- const [Matrix4x3d](#) & **getGlobalTransform** () const
- void **setPivotTransform** (const [Matrix4x3d](#) &transform)

*pivot transform*

- const [Matrix4x3d](#) & **getPivotTransform** () const
- void **setMorphTransform** (const [Vector4f](#) &transform)

*morph transform*

- const [Vector4f](#) & **getMorphTransform** () const
- void **createLocalTransforms** (const [Matrix4x3d](#) &itransform=[Matrix4x3d::identity](#))

*create transforms*

- void **createGlobalTransforms** (const [Matrix4x3d](#) &transform=[Matrix4x3d::identity](#))
- void **setTransform** (const [Vector3d](#) &scale)

*apply transform*

- size\_t **getMemory** () const

*memory usage*

## Friends

- class **Mesh**

## 5.196.1 Detailed Description

[MeshNode](#) class

## 5.197 Tellusim::MeshStream Class Reference

```
#include <format/TellusimMesh.h>
```

## Public Member Functions

- virtual [Mesh::Basis](#) **getBasis** () const
- virtual bool **info** ([Stream](#) &stream, [Mesh](#) &mesh, [Mesh::Flags](#) flags, [Async](#) \*async)
- virtual bool **load** ([Stream](#) &stream, [Mesh](#) &mesh, [Mesh::Flags](#) flags, [Async](#) \*async)
- virtual bool **save** ([Stream](#) &stream, const [Mesh](#) &mesh, [Mesh::Flags](#) flags)



## Static Public Member Functions

- static bool [check](#) (const [String](#) &name, uint32\_t magic=0)  
*mesh stream formats*
- static [String](#) [getLoadFormats](#) ()  
*list of supported formats*
- static [String](#) [getSaveFormats](#) ()

## Protected Types

- enum **Flags** {  
    **FlagNone** = 0,  
    **FlagLoad** = (1 << 0),  
    **FlagSave** = (1 << 1),  
    **FlagLoadSave** = (FlagLoad | FlagSave) }

## Protected Member Functions

- **MeshStream** (Flags flags, const char \*name, uint32\_t magic=0)
- **MeshStream** (Flags flags, const InitializerList< const char \*> &names, uint32\_t magic=0)
- **MeshStream** (Flags flags, const InitializerList< const char \*> &names, const InitializerList< uint32\_t > &magics)

## 5.197.1 Detailed Description

[MeshStream](#) class

## 5.198 Tellusim::MeshTransform Class Reference

```
#include <format/TellusimMesh.h>
```

## Classes

- struct [KeyData](#)  
*transform data*

## Public Types

- using [TranslateKeys](#) = Array< [KeyData](#)< [Vector3d](#) > >  
*translation keys*
- using [RotateKeys](#) = Array< [KeyData](#)< [Quaternionf](#) > >  
*rotation keys*
- using [ScaleKeys](#) = Array< [KeyData](#)< [Vector3f](#) > >  
*scaling keys*
- using [MorphKeys](#) = Array< [KeyData](#)< [Vector4f](#) > >  
*morphing keys*

## Public Member Functions

- void **clear** ()  
*clear transform*
- float64\_t **getMinTime** () const  
*time range*
- float64\_t **getMaxTime** () const
- void **setTransform** (float64\_t time, const **Matrix4x3d** &transform, float32\_t threshold=1e-6f)  
*set transform*
- void **setTranslate** (float64\_t time, const **Vector3d** &translate, float32\_t threshold=1e-6f)
- void **setRotate** (float64\_t time, const **Quaternionf** &rotate, float32\_t threshold=1e-6f)
- void **setScale** (float64\_t time, const **Vector3f** &scale, float32\_t threshold=1e-6f)
- void **setMorph** (float64\_t time, const **Vector4f** &morph, float32\_t threshold=1e-6f)
- **Matrix4x3d** **getTransform** (float64\_t time) const  
*get transform*
- **Vector3d** **getTranslate** (float64\_t time) const
- **Quaternionf** **getRotate** (float64\_t time) const
- **Vector3f** **getScale** (float64\_t time) const
- **Vector4f** **getMorph** (float64\_t time) const
- bool **hasTransformKeys** () const  
*transform keys*
- void **setTranslateKeys** (const **TranslateKeys** &keys, float64\_t scale=1.0)
- **TranslateKeys** **getTranslateKeys** () const
- bool **hasTranslateKeys** () const
- void **setRotateKeys** (const **RotateKeys** &keys)
- **RotateKeys** **getRotateKeys** () const
- bool **hasRotateKeys** () const
- void **setScaleKeys** (const **ScaleKeys** &keys)
- **ScaleKeys** **getScaleKeys** () const
- bool **hasScaleKeys** () const
- void **setMorphKeys** (const **MorphKeys** &keys)
- **MorphKeys** **getMorphKeys** () const
- bool **hasMorphKeys** () const
- void **setTransform** (const **Vector3d** &scale)  
*apply transform*
- void **setTransform** (const **Matrix4x3d** &transform)
- size\_t **getMemory** () const  
*memory usage*

## 5.198.1 Detailed Description

**MeshTransform** class

## 5.199 Tellusim::Mipmap Struct Reference

## Public Member Functions

- **Mipmap** (uint32\_t base)
- **Mipmap** (uint32\_t base, uint32\_t size)

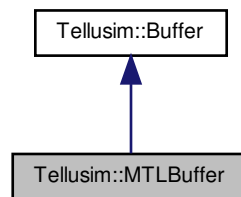
## Public Attributes

- uint32\_t **base** = 0
- uint32\_t **size** = 1

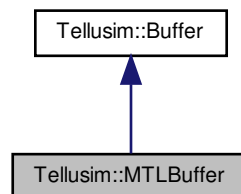
## 5.200 Tellusim::MTLBuffer Class Reference

```
#include <platform/TellusimBuffer.h>
```

Inheritance diagram for Tellusim::MTLBuffer:



Collaboration diagram for Tellusim::MTLBuffer:



## Public Member Functions

- bool **create** (**Flags** flags, void \*buffer)  
*create external buffer*
- void \* **getMTLBuffer** () const
- void \* **getSharedPtr** () const

## Additional Inherited Members

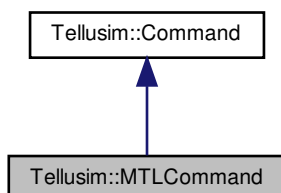
## 5.200.1 Detailed Description

[MTLBuffer](#)

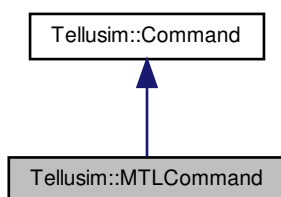
## 5.201 Tellusim::MtlCommand Class Reference

```
#include <platform/TellusimCommand.h>
```

Inheritance diagram for Tellusim::MtlCommand:



Collaboration diagram for Tellusim::MtlCommand:



### Public Member Functions

- void \* [getEncoder](#) () const  
*command context*
- void [flush](#) (void \*encoder, bool enqueue=false)  
*end encoding*
- void **flush** (bool create=false, bool enqueue=false)
- void [update](#) ()  
*update resources*

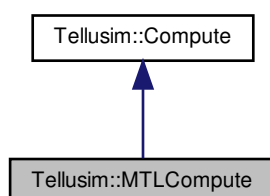
#### 5.201.1 Detailed Description

[MtlCommand](#)

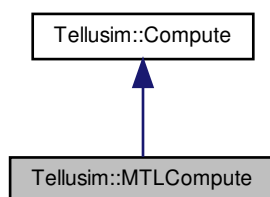
## 5.202 Tellusim::MTLCompute Class Reference

```
#include <platform/TellusimCompute.h>
```

Inheritance diagram for Tellusim::MTLCompute:



Collaboration diagram for Tellusim::MTLCompute:



### Public Member Functions

- void \* [getEncoder](#) () const  
*command context*
- void [flush](#) (void \*encoder, bool enqueue=false)  
*end encoding*
- void **flush** (bool create=false, bool enqueue=false)
- void [update](#) ()  
*update resources*

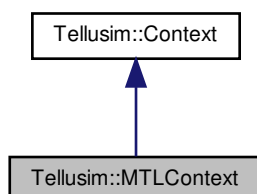
### 5.202.1 Detailed Description

[MTLCompute](#)

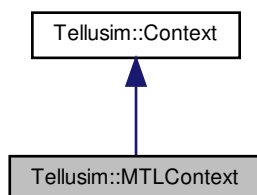
### 5.203 Tellusim::MTLContext Class Reference

```
#include <platform/TellusimContext.h>
```

Inheritance diagram for Tellusim::MTLContext:



Collaboration diagram for Tellusim::MTLContext:



#### Public Member Functions

- bool **create** (void \*device, void \*queue)  
*create context*
- void \* **getDevice** () const  
*current device*
- void \* **getQueue** () const
- void \* **getCommand** () const
- void \* **getEncoder** () const  
*command encoder*
- void \* **getRenderEncoder** (void \*descriptor) const
- void \* **getComputeEncoder** () const
- void \* **getTracingEncoder** () const
- void \* **getBlitEncoder** () const
- void **endEncoder** () const

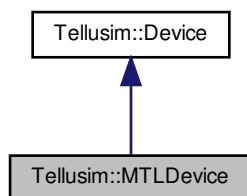
## 5.203.1 Detailed Description

[MTLContext](#)

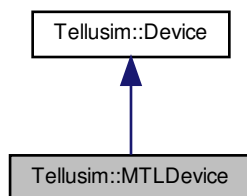
## 5.204 Tellusim::MTLDevice Class Reference

```
#include <platform/TellusimDevice.h>
```

Inheritance diagram for Tellusim::MTLDevice:



Collaboration diagram for Tellusim::MTLDevice:



## Public Member Functions

- **MTLDevice** ([Context](#) &context)
- **MTLDevice** ([Surface](#) &surface)
- **MTLDevice** ([Window](#) &window)
- void \* [getMTLDevice](#) () const  
    *command context*
- void \* **getQueue** () const
- void \* **getCommand** () const
- void \* [getEncoder](#) () const  
    *command encoder*
- void \* **getRenderEncoder** (void \*descriptor) const
- void \* **getComputeEncoder** () const
- void \* **getTracingEncoder** () const
- void \* **getBlitEncoder** () const
- void **endEncoder** () const

#### 5.204.1 Detailed Description

[MTLDevice](#)

### 5.205 Tellusim::MTLTracing::MTLInstance Struct Reference

tracing instance

```
#include <platform/TellusimTracing.h>
```

#### Public Attributes

- float32\_t **transform** [12]
- uint32\_t **options**
- uint32\_t **mask**
- uint32\_t **offset**
- uint32\_t **index**

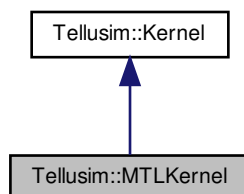
#### 5.205.1 Detailed Description

tracing instance

### 5.206 Tellusim::MTLKernel Class Reference

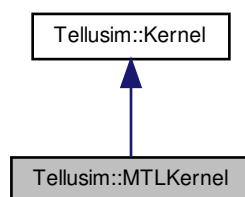
```
#include <platform/TellusimKernel.h>
```

Inheritance diagram for Tellusim::MTLKernel:





Collaboration diagram for Tellusim::MTLKernel:



#### Public Member Functions

- void **setIndirect** (bool enabled)  
*indirect command buffer*
- bool **isIndirect** () const
- void \* **getComputeFunction** () const

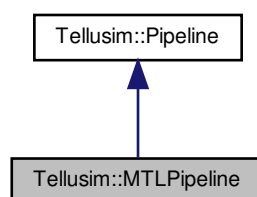
#### 5.206.1 Detailed Description

[MTLKernel](#)

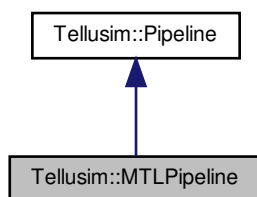
## 5.207 Tellusim::MTLPipeline Class Reference

```
#include <platform/TellusimPipeline.h>
```

Inheritance diagram for Tellusim::MTLPipeline:



Collaboration diagram for Tellusim::MTLPipeline:



#### Public Member Functions

- void **setIndirect** (bool enabled)  
*indirect command buffer*
- bool **isIndirect** () const
- void \* **getVertexFunction** () const
- void \* **getFragmentFunction** () const

#### Additional Inherited Members

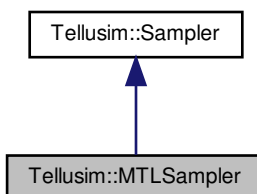
#### 5.207.1 Detailed Description

#### MTLPipeline

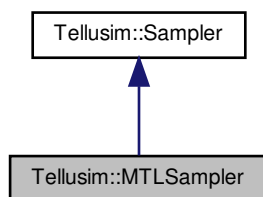
#### 5.208 Tellusim::MTLSampler Class Reference

```
#include <platform/TellusimSampler.h>
```

Inheritance diagram for Tellusim::MTLSampler:



Collaboration diagram for Tellusim::MTLSampler:



#### Public Member Functions

- void [setIndirect](#) (bool enabled)  
*indirect command buffer*
- bool **isIndirect** () const

#### Additional Inherited Members

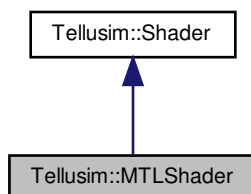
##### 5.208.1 Detailed Description

#### [MTLSampler](#)

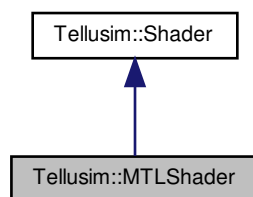
## 5.209 Tellusim::MTLShader Class Reference

```
#include <platform/TellusimShader.h>
```

Inheritance diagram for Tellusim::MTLShader:



Collaboration diagram for Tellusim::MTLShader:



#### Public Member Functions

- void [setIndirect](#) (bool enabled)  
*indirect command buffer*
- bool **isIndirect** () const
- void \* **getLibrary** () const
- void \* **getFunction** () const

#### Additional Inherited Members

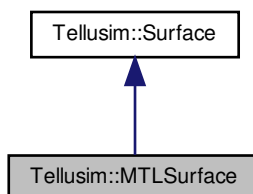
##### 5.209.1 Detailed Description

#### [MTLShader](#)

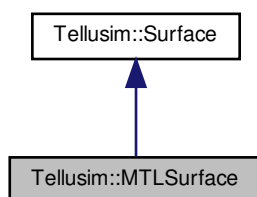
### 5.210 Tellusim::MTLSurface Class Reference

```
#include <platform/TellusimSurface.h>
```

Inheritance diagram for Tellusim::MTLSurface:



Collaboration diagram for Tellusim::MTLSurface:



#### Public Member Functions

- **MTLSurface** ([MTLContext](#) &context)
- void \* [getDevice](#) () const  
*current device*
- void \* **getQueue** () const
- void \* **getCommand** () const
- void [setDescriptor](#) (void \*descriptor)  
*render pass descriptor*
- void \* **getDescriptor** () const
- uint32\_t [getColorPixelFormat](#) () const  
*surface formats*
- uint32\_t **getDepthPixelFormat** () const

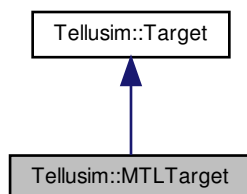
#### 5.210.1 Detailed Description

[MTLSurface](#)

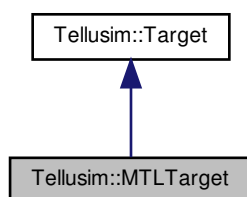
### 5.211 Tellusim::MTLTarget Class Reference

```
#include <platform/TellusimTarget.h>
```

Inheritance diagram for Tellusim::MTLTarget:



Collaboration diagram for Tellusim::MTLTarget:



#### Public Member Functions

- void \* **getDescriptor** () const

#### Additional Inherited Members

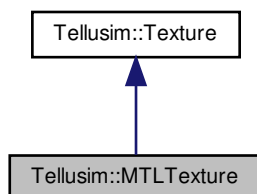
##### 5.211.1 Detailed Description

[MTLTarget](#)

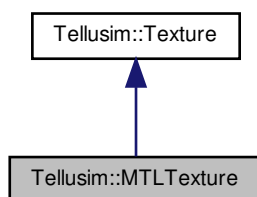
## 5.212 Tellusim::MTLTexture Class Reference

```
#include <platform/TellusimTexture.h>
```

Inheritance diagram for Tellusim::MTLTexture:



Collaboration diagram for Tellusim::MTLTexture:



#### Public Member Functions

- bool **create** (void \*texture, [Flags](#) flags=DefaultFlags, Format format=FormatUnknown)  
*create external texture*
- uint32\_t **getPixelFormat** () const
- uint32\_t **getTextureType** () const
- void \* **getMTLTexture** () const
- void \* **getMTLBuffer** () const
- void \* **getSharedPtr** () const

#### Additional Inherited Members

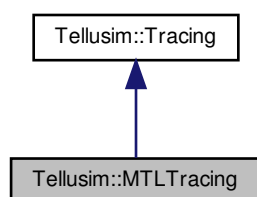
##### 5.212.1 Detailed Description

#### [MTLTexture](#)

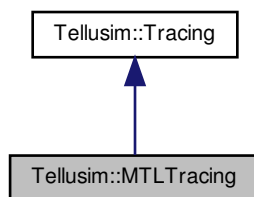
## 5.213 Tellusim::MTLTracing Class Reference

```
#include <platform/TellusimTracing.h>
```

Inheritance diagram for Tellusim::MTLTracing:



Collaboration diagram for Tellusim::MTLTracing:



## Classes

- struct [MTLInstance](#)  
*tracing instance*

## Public Member Functions

- void \* **getGeometryDesc** (uint32\_t index) const
- void \* **getPrimitiveDesc** () const
- void \* **getInstanceDesc** () const
- void \* **getAccelerationStructure** () const

## Additional Inherited Members

### 5.213.1 Detailed Description

#### [MTLTracing](#)

### 5.214 Tellusim::Spatial::Node< Type, Size > Struct Template Reference

#### [Spatial Node](#).

```
#include <geometry/TellusimSpatial.h>
```

## Public Types

- enum { **Axes** = Size }
- using **Bound** = Type



## Public Attributes

- Bound **bound**
- uint32\_t **left**
- uint32\_t **right**
- uint32\_t **parent**
- uint32\_t **spatial**

## 5.214.1 Detailed Description

```
template<class Type, uint32_t Size>
struct Tellusim::Spatial::Node< Type, Size >
```

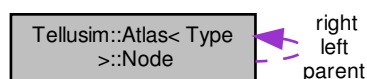
[Spatial Node.](#)

## 5.215 Tellusim::Atlas&lt; Type &gt;::Node Struct Reference

[Atlas Node.](#)

```
#include <geometry/TellusimAtlas.h>
```

Collaboration diagram for Tellusim::Atlas< Type >::Node:



## Public Attributes

- Bound **bound**
- [Node](#) \* **left** = nullptr
- [Node](#) \* **right** = nullptr
- [Node](#) \* **parent** = nullptr
- uint32\_t **axis** = Maxu32

## 5.215.1 Detailed Description

```
template<class Type>
struct Tellusim::Atlas< Type >::Node
```

[Atlas Node.](#)

## 5.216 Tellusim::SpatialTree::Node Struct Reference

### Public Attributes

- float32\_t **bound\_min** [3]
- float32\_t **is\_enabled**
- float32\_t **bound\_max** [3]
- uint32\_t **data**
- uint32\_t **left**
- uint32\_t **right**
- uint32\_t **parent**
- uint32\_t **spatial**

## 5.217 Tellusim::Origin Struct Reference

```
#include <TellusimTypes.h>
```

### Public Member Functions

- **Origin** (uint32\_t x, uint32\_t y)
- **Origin** (uint32\_t x, uint32\_t y, uint32\_t z)

### Public Attributes

- uint32\_t **x** = 0
- uint32\_t **y** = 0
- uint32\_t **z** = 0

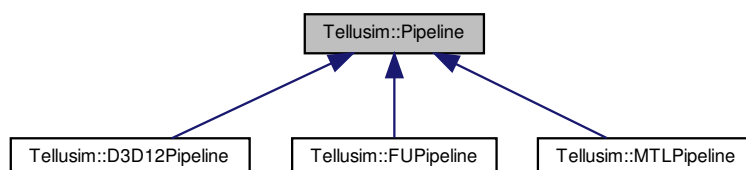
### 5.217.1 Detailed Description

[Origin](#)

## 5.218 Tellusim::Pipeline Class Reference

```
#include <platform/TellusimPipeline.h>
```

Inheritance diagram for Tellusim::Pipeline:



## Public Types

- enum [Primitive](#) {  
    **PrimitivePoint** = 0,  
    **PrimitivePointPatch**,  
    **PrimitiveLine**,  
    **PrimitiveLineAdj**,  
    **PrimitiveLineStrip**,  
    **PrimitiveLinePatch**,  
    **PrimitiveTriangle**,  
    **PrimitiveTriangleAdj**,  
    **PrimitiveTriangleStrip**,  
    **PrimitiveTrianglePatch**,  
    **PrimitiveQuadrilateralPatch**,  
    **NumPrimitiveTypes** }

*Primitive types.*

- enum [Attribute](#) {  
    **AttributePosition** = 0,  
    **AttributeBasis**,  
    **AttributeNormal**,  
    **AttributeTangent**,  
    **AttributeBinormal**,  
    **AttributeTexCoord**,  
    **AttributeWeights**,  
    **AttributeJoints**,  
    **AttributeColor**,  
    **NumAttributeTypes** }

*Attribute types.*

- enum [FillMode](#) {  
    **FillModeLine** = 0,  
    **FillModeSolid**,  
    **NumFillModes** }

*Filling modes.*

- enum [CullMode](#) {  
    **CullModeNone** = 0,  
    **CullModeBack**,  
    **CullModeFront**,  
    **NumCullModes** }

*Culling modes.*

- enum [FrontMode](#) {  
    **FrontModeCCW** = 0,  
    **FrontModeCW**,  
    **NumFrontModes** }

*Front modes.*

- enum [BlendOp](#) {  
    **BlendOpAdd** = 0,  
    **BlendOpSub**,  
    **BlendOpMin**,  
    **BlendOpMax**,  
    **NumBlendOperations** }

*Blending operations.*

- enum [BlendFunc](#) {  
    **BlendFuncNone** = 0,  
    **BlendFuncZero**,  
    **BlendFuncOne**,  
    **BlendFuncSrcColor**,  
    **BlendFuncSrcAlpha**,

```

BlendFuncSrc1Color,
BlendFuncSrc1Alpha,
BlendFuncDestColor,
BlendFuncDestAlpha,
BlendFuncFactorColor,
BlendFuncFactorAlpha,
BlendFuncInvSrcColor,
BlendFuncInvSrcAlpha,
BlendFuncInvSrc1Color,
BlendFuncInvSrc1Alpha,
BlendFuncInvDestColor,
BlendFuncInvDestAlpha,
BlendFuncInvFactorColor,
BlendFuncInvFactorAlpha,
NumBlendFunctions }

```

*Blending functions.*

- enum [ColorMask](#) {  
**ColorMaskNone** = 0,  
**ColorMaskR** = (1 << 0),  
**ColorMaskG** = (1 << 1),  
**ColorMaskB** = (1 << 2),  
**ColorMaskA** = (1 << 3),  
**ColorMaskUnknown** = (1 << 4),  
**ColorMaskRGB** = (ColorMaskR | ColorMaskG | ColorMaskB),  
**ColorMaskAll** = (ColorMaskRGB | ColorMaskA) }

*Color masks.*

- enum [DepthMask](#) {  
**DepthMaskNone** = 0,  
**DepthMaskRead**,  
**DepthMaskWrite**,  
**NumDepthMasks** }

*Depth masks.*

- enum [DepthFunc](#) {  
**DepthFuncNone** = 0,  
**DepthFuncNever**,  
**DepthFuncAlways**,  
**DepthFuncEqual**,  
**DepthFuncLess**,  
**DepthFuncGreater**,  
**DepthFuncNotEqual**,  
**DepthFuncLessEqual**,  
**DepthFuncGreaterEqual**,  
**NumDepthFunctions** }

*Depth functions.*

- enum [StencilOp](#) {  
**StencilOpKeep** = 0,  
**StencilOpInvert**,  
**StencilOpReplace**,  
**StencilOpIncrWrap**,  
**StencilOpDecrWrap**,  
**StencilOpIncrSat**,  
**StencilOpDecrSat**,  
**NumStencilOperations** }

*Stencil operations.*

- enum [StencilFunc](#) {  
**StencilFuncNone** = 0,  
**StencilFuncNever**,

```

StencilFuncAlways,
StencilFuncEqual,
StencilFuncLess,
StencilFuncGreater,
StencilFuncNotEqual,
StencilFuncLessEqual,
StencilFuncGreaterEqual,
NumStencilFunctions }

```

*Stencil functions.*

## Public Member Functions

- Platform [getPlatform](#) () const  
*pipeline platform*
- const char \* [getPlatformName](#) () const
- uint32\_t [getIndex](#) () const  
*pipeline device index*
- void [clear](#) ()  
*clear pipeline*
- bool [isCreated](#) () const  
*check pipeline*
- void [setName](#) (const char \*name)  
*pipeline name*
- [String](#) [getName](#) () const
- bool [create](#) ()  
*create pipeline*
- void [setParameters](#) (const [Pipeline](#) &pipeline)  
*pipeline parameters*
- bool [saveState](#) ([Stream](#) &stream) const
- void [addShader](#) ([Shader](#) &shader, bool owner=false)  
*shader pointers*
- [Shader](#) [getVertexShader](#) () const
- [Shader](#) [getControlShader](#) () const
- [Shader](#) [getEvaluateShader](#) () const
- [Shader](#) [getGeometryShader](#) () const
- [Shader](#) [getFragmentShader](#) () const
- [Shader](#) [getTaskShader](#) () const
- [Shader](#) [getMeshShader](#) () const
- bool [loadShader](#) ([Shader::Type](#) type, const char \*name, const char \*format,...) 1(4  
*load shaders*
- bool [loadShaderGLSL](#) ([Shader::Type](#) type, const char \*name, const char \*format,...) 1(4
- bool [loadShader](#) ([Shader::Type](#) type, const char \*name, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool [loadShaderGLSL](#) ([Shader::Type](#) type, const char \*name, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool [loadShaderSPIRV](#) ([Shader::Type](#) type, const char \*name)
- bool [createShader](#) ([Shader::Type](#) type, const char \*src, const char \*format,...) 1(4  
*create shaders*
- bool [createShaderGLSL](#) ([Shader::Type](#) type, const char \*src, const char \*format,...) 1(4
- bool [createShader](#) ([Shader::Type](#) type, const char \*src, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool [createShaderGLSL](#) ([Shader::Type](#) type, const char \*src, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)

- bool **createShaderSPIRV** (Shader::Type type, const Array< uint32\_t > &data)
- uint32\_t **addSampler** (Shader::Mask mask)
  - sampler parameters*
  - uint32\_t **getNumSamplers** () const
  - Pipeline & **setSamplerOffset** (uint32\_t offset)
  - uint32\_t **getSamplerOffset** () const
  - Pipeline & **setSamplerMask** (uint32\_t index, Shader::Mask mask)
  - Shader::Mask **getSamplerMask** (uint32\_t index) const
  - Pipeline & **setSamplerMasks** (uint32\_t index, uint32\_t num, Shader::Mask mask, bool array=false)
  - Shader::Mask **getSamplerMasks** (uint32\_t index, uint32\_t num) const
  - Pipeline & **setSamplerArray** (uint32\_t index, uint32\_t num, bool array)
  - uint32\_t **getSamplerArray** (uint32\_t index) const
  - uint32\_t **addTexture** (Shader::Mask mask)
    - texture parameters*
    - uint32\_t **getNumTextures** () const
    - Pipeline & **setTextureOffset** (uint32\_t offset)
    - uint32\_t **getTextureOffset** () const
    - Pipeline & **setTextureMask** (uint32\_t index, Shader::Mask mask)
    - Shader::Mask **getTextureMask** (uint32\_t index) const
    - Pipeline & **setTextureMasks** (uint32\_t index, uint32\_t num, Shader::Mask mask, bool array=false)
    - Shader::Mask **getTextureMasks** (uint32\_t index, uint32\_t num) const
    - Pipeline & **setTextureArray** (uint32\_t index, uint32\_t num, bool array)
    - uint32\_t **getTextureArray** (uint32\_t index) const
    - uint32\_t **addSurface** (Shader::Mask mask)
      - surface parameters*
      - uint32\_t **getNumSurfaces** () const
      - Pipeline & **setSurfaceOffset** (uint32\_t offset)
      - uint32\_t **getSurfaceOffset** () const
      - Pipeline & **setSurfaceMask** (uint32\_t index, Shader::Mask mask)
      - Shader::Mask **getSurfaceMask** (uint32\_t index) const
      - Pipeline & **setSurfaceMasks** (uint32\_t index, uint32\_t num, Shader::Mask mask, bool array=false)
      - Shader::Mask **getSurfaceMasks** (uint32\_t index, uint32\_t num) const
      - Pipeline & **setSurfaceArray** (uint32\_t index, uint32\_t num, bool array)
      - uint32\_t **getSurfaceArray** (uint32\_t index) const
      - uint32\_t **addUniform** (Shader::Mask mask, BindFlags flags=BindFlagNone)
        - uniform parameters*
        - uint32\_t **getNumUniforms** () const
        - Pipeline & **setUniformOffset** (uint32\_t offset)
        - uint32\_t **getUniformOffset** () const
        - Pipeline & **setUniformMask** (uint32\_t index, Shader::Mask mask, BindFlags flags=BindFlagNone)
        - Shader::Mask **getUniformMask** (uint32\_t index) const
        - Pipeline & **setUniformMasks** (uint32\_t index, uint32\_t num, Shader::Mask mask, BindFlags flags=BindFlagNone)
        - Shader::Mask **getUniformMasks** (uint32\_t index, uint32\_t num) const
        - Pipeline & **setUniformFlags** (uint32\_t index, BindFlags flags)
        - BindFlags **getUniformFlags** (uint32\_t index) const
        - uint32\_t **addStorage** (Shader::Mask mask, BindFlags flags=BindFlagNone)
          - storage parameters*
          - uint32\_t **getNumStorages** () const
          - Pipeline & **setStorageOffset** (uint32\_t offset)
          - uint32\_t **getStorageOffset** () const
          - Pipeline & **setStorageMask** (uint32\_t index, Shader::Mask mask, BindFlags flags=BindFlagNone)
          - Shader::Mask **getStorageMask** (uint32\_t index) const

- [Pipeline](#) & **setStorageMasks** (uint32\_t index, uint32\_t num, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
- [Shader::Mask](#) **getStorageMasks** (uint32\_t index, uint32\_t num) const
- [Pipeline](#) & **setStorageFlags** (uint32\_t index, BindFlags flags)
- BindFlags **getStorageFlags** (uint32\_t index) const
- uint32\_t **addTracing** ([Shader::Mask](#) mask)
- tracing parameters*
- uint32\_t **getNumTracings** () const
- [Pipeline](#) & **setTracingOffset** (uint32\_t offset)
- uint32\_t **getTracingOffset** () const
- [Pipeline](#) & **setTracingMask** (uint32\_t index, [Shader::Mask](#) mask)
- [Shader::Mask](#) **getTracingMask** (uint32\_t index) const
- [Pipeline](#) & **setTracingMasks** (uint32\_t index, uint32\_t num, [Shader::Mask](#) mask)
- [Shader::Mask](#) **getTracingMasks** (uint32\_t index, uint32\_t num) const
- uint32\_t **addTexel** ([Shader::Mask](#) mask)
- texel parameters*
- uint32\_t **getNumTexels** () const
- [Pipeline](#) & **setTexelOffset** (uint32\_t offset)
- uint32\_t **getTexelOffset** () const
- [Pipeline](#) & **setTexelMask** (uint32\_t index, [Shader::Mask](#) mask)
- [Shader::Mask](#) **getTexelMask** (uint32\_t index) const
- [Pipeline](#) & **setTexelMasks** (uint32\_t index, uint32\_t num, [Shader::Mask](#) mask)
- [Shader::Mask](#) **getTexelMasks** (uint32\_t index, uint32\_t num) const
- uint32\_t **addTable** (TableType type, uint32\_t size, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
- table parameters*
- uint32\_t **getNumTables** () const
- [Pipeline](#) & **setTableOffset** (uint32\_t offset)
- uint32\_t **getTableOffset** () const
- [Pipeline](#) & **setTableType** (uint32\_t index, TableType type, uint32\_t size, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
- TableType **getTableType** (uint32\_t index) const
- uint32\_t **getTableSize** (uint32\_t index) const
- [Pipeline](#) & **setTableMask** (uint32\_t index, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
- [Shader::Mask](#) **getTableMask** (uint32\_t index) const
- [Pipeline](#) & **setTableFlags** (uint32\_t index, BindFlags flags)
- BindFlags **getTableFlags** (uint32\_t index) const
- uint32\_t **getNumVertices** () const
- vertex parameters*
- uint32\_t **getVertexStride** (uint32\_t index) const
- uint32\_t **getVertexRate** (uint32\_t index) const
- uint32\_t **addAttribute** ([Attribute](#) attribute, Format format, uint32\_t vertex, size\_t offset, size\_t stride, uint32\_t rate=0)
- vertex attributes*
- [Pipeline](#) & **setAttribute** (uint32\_t index, [Attribute](#) attribute, Format format, uint32\_t vertex, size\_t offset, size\_t stride, uint32\_t rate=0)
- [Pipeline](#) & **setAttributeType** (uint32\_t index, [Attribute](#) attribute)
- [Pipeline](#) & **setAttributeFormat** (uint32\_t index, Format format)
- [Pipeline](#) & **setAttributeVertex** (uint32\_t index, uint32\_t vertex)
- [Pipeline](#) & **setAttributeOffset** (uint32\_t index, size\_t offset)
- [Pipeline](#) & **setAttributeStride** (uint32\_t index, size\_t stride)
- [Pipeline](#) & **setAttributeRate** (uint32\_t index, uint32\_t rate)
- uint32\_t **getNumAttributes** () const
- [Attribute](#) **getAttributeType** (uint32\_t index) const
- Format **getAttributeFormat** (uint32\_t index) const

- uint32\_t **getAttributeVertex** (uint32\_t index) const
- uint32\_t **getAttributeOffset** (uint32\_t index) const
- uint32\_t **getAttributeStride** (uint32\_t index) const
- uint32\_t **getAttributeRate** (uint32\_t index) const
- void **setPrimitive** (**Primitive** primitive)
- rasterization parameters*
- **Primitive** **getPrimitive** () const
- void **setFillMode** (**FillMode** mode)
- **FillMode** **getFillMode** () const
- void **setCullMode** (**CullMode** mode)
- **CullMode** **getCullMode** () const
- void **setFrontMode** (**FrontMode** mode)
- **FrontMode** **getFrontMode** () const
- void **setDepthBias** (float32\_t bias, float32\_t slope, float32\_t clamp=0.0f)
- float32\_t **getDepthBias** () const
- float32\_t **getDepthSlope** () const
- float32\_t **getDepthClamp** () const
- void **setMultisample** (uint32\_t multisample)
- uint32\_t **getMultisample** () const
- void **setSampleMask** (uint32\_t sample\_mask)
- uint32\_t **getSampleMask** () const
- void **setDepthClip** (bool enabled)
- bool **getDepthClip** () const
- void **setDepthReplace** (bool enabled)
- bool **getDepthReplace** () const
- void **setScissorTest** (bool enabled)
- bool **getScissorTest** () const
- void **setRasterDiscard** (bool enabled)
- bool **getRasterDiscard** () const
- void **setSampleShading** (bool enabled)
- bool **getSampleShading** () const
- void **setAlphaToCoverage** (bool enabled)
- bool **getAlphaToCoverage** () const
- void **setMultisampleRaster** (bool enabled)
- bool **getMultisampleRaster** () const
- void **setConservativeRaster** (bool enabled)
- bool **getConservativeRaster** () const
- void **setNumViewports** (uint32\_t num\_viewports)
- uint32\_t **getNumTargets** () const
- uint32\_t **getNumViewports** () const
- void **setNumClipDistances** (uint32\_t num\_distances)
- uint32\_t **getNumClipDistances** () const
- void **setBlend** (**BlendOp** op, **BlendFunc** src, **BlendFunc** dest)
- blending parameters*
- void **setBlendColor** (**BlendOp** op, **BlendFunc** src, **BlendFunc** dest)
- void **setBlendAlpha** (**BlendOp** op, **BlendFunc** src, **BlendFunc** dest)
- void **setBlend** (uint32\_t index, **BlendOp** op, **BlendFunc** src, **BlendFunc** dest)
- void **setBlendColor** (uint32\_t index, **BlendOp** op, **BlendFunc** src, **BlendFunc** dest)
- void **setBlendAlpha** (uint32\_t index, **BlendOp** op, **BlendFunc** src, **BlendFunc** dest)
- **BlendOp** **getBlendColorOp** (uint32\_t index) const
- **BlendOp** **getBlendAlphaOp** (uint32\_t index) const
- **BlendFunc** **getBlendSrcColorFunc** (uint32\_t index) const
- **BlendFunc** **getBlendSrcAlphaFunc** (uint32\_t index) const
- **BlendFunc** **getBlendDestColorFunc** (uint32\_t index) const



- [BlendFunc](#) **getBlendDestAlphaFunc** (uint32\_t index) const
- void **setColorMask** ([ColorMask](#) mask)
  - color parameters*
- void **setColorMask** (uint32\_t index, [ColorMask](#) mask)
- void **setColorFormat** (uint32\_t index, Format format)
- void **setColorFormat** (Format format, uint32\_t num=1)
- [ColorMask](#) **getColorMask** (uint32\_t index) const
- Format **getColorFormat** (uint32\_t index) const
- void **setDepthMask** ([DepthMask](#) mask)
  - depth parameters*
- void **setDepthFunc** ([DepthFunc](#) func)
- void **setDepthFormat** (Format format)
- [DepthMask](#) **getDepthMask** () const
- [DepthFunc](#) **getDepthFunc** () const
- Format **getDepthFormat** () const
- void **setStencilMask** (uint32\_t mask)
  - stencil parameters*
- void **setStencilBackMask** (uint32\_t mask)
- void **setStencilFrontMask** (uint32\_t mask)
- void **setStencilFunc** ([StencilFunc](#) func, [StencilOp](#) dpass\_op)
- void **setStencilBackFunc** ([StencilFunc](#) func, [StencilOp](#) dpass\_op)
- void **setStencilFrontFunc** ([StencilFunc](#) func, [StencilOp](#) dpass\_op)
- void **setStencilFunc** ([StencilFunc](#) func, [StencilOp](#) fail\_op, [StencilOp](#) dfail\_op, [StencilOp](#) dpass\_op)
- void **setStencilBackFunc** ([StencilFunc](#) func, [StencilOp](#) fail\_op, [StencilOp](#) dfail\_op, [StencilOp](#) dpass\_op)
- void **setStencilFrontFunc** ([StencilFunc](#) func, [StencilOp](#) fail\_op, [StencilOp](#) dfail\_op, [StencilOp](#) dpass\_op)
- uint32\_t **getStencilBackMask** () const
- [StencilFunc](#) **getStencilBackFunc** () const
- [StencilOp](#) **getStencilBackFailOp** () const
- [StencilOp](#) **getStencilBackDepthFailOp** () const
- [StencilOp](#) **getStencilBackDepthPassOp** () const
- uint32\_t **getStencilFrontMask** () const
- [StencilFunc](#) **getStencilFrontFunc** () const
- [StencilOp](#) **getStencilFrontFailOp** () const
- [StencilOp](#) **getStencilFrontDepthFailOp** () const
- [StencilOp](#) **getStencilFrontDepthPassOp** () const

#### 5.218.1 Detailed Description

[Pipeline](#) class

## 5.219 Tellusim::Polygon< Capacity > Struct Template Reference

```
#include <geometry/TellusimPolygon.h>
```

## Static Public Member Functions

- `template<class Type >`  
`static Vector3< Type > normal (const Vector3< Type > *vertices, uint32_t num_vertices)`  
*3D polygon normal*
- `template<class Type , class Index >`  
`static uint32_t triangulate (const Vector3< Type > *vertices, uint32_t num_vertices, Index *indices)`  
*triangulate 3D polygon*
- `template<class Type , class Index >`  
`static uint32_t triangulate (const Vector3< Type > *vertices, uint32_t num_vertices, const Vector3< Type > &normal, Index *indices)`  
*triangulate 3D polygon with specified normal*
- `template<class Type , class Index >`  
`static uint32_t triangulate (const Vector2< Type > *vertices, uint32_t num_vertices, Index *indices)`  
*triangulate 2D polygon*

### 5.219.1 Detailed Description

```
template<uint32_t Capacity = 256>
struct Tellusim::Polygon< Capacity >
```

[Polygon](#) utils

### 5.219.2 Member Function Documentation

#### 5.219.2.1 normal()

```
template<uint32_t Capacity = 256>
template<class Type >
static Vector3<Type> Tellusim::Polygon< Capacity >::normal (
    const Vector3< Type > * vertices,
    uint32_t num_vertices ) [inline], [static]
```

3D polygon normal

vertex bounds

maximum axis

maximum area

#### 5.219.2.2 triangulate() [1/3]

```
template<uint32_t Capacity = 256>
template<class Type , class Index >
static uint32_t Tellusim::Polygon< Capacity >::triangulate (
    const Vector3< Type > * vertices,
    uint32_t num_vertices,
    Index * indices ) [inline], [static]
```

triangulate 3D polygon

polygon normal

triangulate polygon

## 5.219.2.3 triangulate() [2/3]

```
template<uint32_t Capacity = 256>
template<class Type , class Index >
static uint32_t Tellusim::Polygon< Capacity >::triangulate (
    const Vector3< Type > * vertices,
    uint32_t num_vertices,
    const Vector3< Type > & normal,
    Index * indices ) [inline], [static]
```

triangulate 3D polygon with specified normal

polygon normal basis

project polygon vertices

triangulate polygon

## 5.219.2.4 triangulate() [3/3]

```
template<uint32_t Capacity = 256>
template<class Type , class Index >
static uint32_t Tellusim::Polygon< Capacity >::triangulate (
    const Vector2< Type > * vertices,
    uint32_t num_vertices,
    Index * indices ) [inline], [static]
```

triangulate 2D polygon

triangle polygon

signed polygon area

polygon winding

vertex angles

triangulate polygon

find next vertex

next vertex

update angles

next triangle

## 5.220 Tellusim::PrefixScan Class Reference

```
#include <parallel/TellusimPrefixScan.h>
```

## Classes

- struct [DispatchParameters](#)

## Public Types

- enum [Mode](#) {  
**ModeSingle** = 0,  
**ModeMultiple**,  
**NumModes** }  
*Scan modes.*
- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagSingle** = (1 << ModeSingle),  
**FlagMultiple** = (1 << ModeMultiple),  
**FlagIndirect** = (1 << (NumModes + 0)),  
**FlagRepeat** = (1 << (NumModes + 1)),  
**FlagsAll** = (FlagSingle | FlagMultiple | FlagIndirect) }  
*Scan flags.*

## Public Member Functions

- void [clear](#) ()  
*clear scan*
- bool [isCreated](#) ([Flags](#) flags) const  
*check scan*
- uint32\_t [getGroupSize](#) () const  
*scan parameters*
- uint32\_t [getScanElements](#) () const
- uint32\_t [getMaxElements](#) () const
- uint32\_t [getMaxRegions](#) () const
- bool [create](#) (const [Device](#) &device, [Mode](#) mode, uint32\_t groups=256, uint32\_t regions=1, [Async](#) \*async=NULLPTR)
- bool [create](#) (const [Device](#) &device, [Flags](#) flags, uint32\_t groups=256, uint32\_t regions=1, [Async](#) \*async=NULLPTR)
- bool [dispatch](#) ([Compute](#) &compute, [Buffer](#) &data, uint32\_t offset, uint32\_t size)
- bool [dispatch](#) ([Compute](#) &compute, [Buffer](#) &data, uint32\_t count, const uint32\_t \*offsets, const uint32\_t \*sizes, [Flags](#) flags=FlagNone)
- bool [dispatchIndirect](#) ([Compute](#) &compute, [Buffer](#) &data, [Buffer](#) &dispatch, uint32\_t offset, [Flags](#) flags=FlagNone, uint32\_t max\_size=Maxu32)
- bool [dispatchIndirect](#) ([Compute](#) &compute, [Buffer](#) &data, uint32\_t count, [Buffer](#) &dispatch, uint32\_t offset, [Flags](#) flags=FlagNone, uint32\_t max\_size=Maxu32)
- bool [dispatchIndirect](#) ([Compute](#) &compute, [Buffer](#) &data, [Buffer](#) &count, [Buffer](#) &dispatch, uint32\_t count\_offset, uint32\_t dispatch\_offset, [Flags](#) flags=FlagNone, uint32\_t max\_size=Maxu32)

## 5.220.1 Detailed Description

[PrefixScan](#) class

## 5.220.2 Member Function Documentation

## 5.220.2.1 create()

```
bool Tellusim::PrefixScan::create (
    const Device & device,
    Mode mode,
    uint32_t groups = 256,
    uint32_t regions = 1,
    Async * async = nullptr )
```

create prefix scan

## Parameters

|                |                                     |
|----------------|-------------------------------------|
| <i>groups</i>  | Prefix scan group size.             |
| <i>regions</i> | Maximum number of multiple regions. |

## 5.220.2.2 dispatch() [1/2]

```
bool Tellusim::PrefixScan::dispatch (
    Compute & compute,
    Buffer & data,
    uint32_t offset,
    uint32_t size )
```

dispatch inplace single prefix scan

## Parameters

|               |                                      |
|---------------|--------------------------------------|
| <i>data</i>   | Buffer of uint32_t elements to scan. |
| <i>offset</i> | Elements offset index (4 aligned).   |
| <i>size</i>   | Number of uint32_t elements to scan. |

## 5.220.2.3 dispatch() [2/2]

```
bool Tellusim::PrefixScan::dispatch (
    Compute & compute,
    Buffer & data,
    uint32_t count,
    const uint32_t * offsets,
    const uint32_t * sizes,
    Flags flags = FlagNone )
```

dispatch inplace multiple prefix scans

## Parameters

|                |                                      |
|----------------|--------------------------------------|
| <i>data</i>    | Buffer of uint32_t elements to scan. |
| <i>count</i>   | Number of regions to scan.           |
| <i>offsets</i> | Elements offset index (4 aligned).   |
| <i>sizes</i>   | Number of uint32_t elements to scan. |

#### 5.220.2.4 `dispatchIndirect()` [1/3]

```
bool Tellusim::PrefixScan::dispatchIndirect (
    Compute & compute,
    Buffer & data,
    Buffer & dispatch,
    uint32_t offset,
    Flags flags = FlagNone,
    uint32_t max_size = Maxu32 )
```

dispatch inplace single indirect prefix scan

##### Parameters

|                 |                                           |
|-----------------|-------------------------------------------|
| <i>data</i>     | Buffer of uint32_t elements to scan.      |
| <i>dispatch</i> | Dispatch indirect buffer.                 |
| <i>offset</i>   | Dispatch indirect buffer offset.          |
| <i>max_size</i> | Maximum number of elements to scan.       |
| <i>repeat</i>   | Repeat dispatch with the last parameters. |

#### 5.220.2.5 `dispatchIndirect()` [2/3]

```
bool Tellusim::PrefixScan::dispatchIndirect (
    Compute & compute,
    Buffer & data,
    uint32_t count,
    Buffer & dispatch,
    uint32_t offset,
    Flags flags = FlagNone,
    uint32_t max_size = Maxu32 )
```

dispatch inplace multiple indirect prefix scans

##### Parameters

|                 |                                           |
|-----------------|-------------------------------------------|
| <i>data</i>     | Buffer of uint32_t elements to scan.      |
| <i>count</i>    | Number of regions to scan.                |
| <i>dispatch</i> | Dispatch indirect buffer.                 |
| <i>offset</i>   | Dispatch indirect buffer offset.          |
| <i>max_size</i> | Maximum number of elements to scan.       |
| <i>repeat</i>   | Repeat dispatch with the last parameters. |

#### 5.220.2.6 `dispatchIndirect()` [3/3]

```
bool Tellusim::PrefixScan::dispatchIndirect (
    Compute & compute,
```

```
Buffer & data,  
Buffer & count,  
Buffer & dispatch,  
uint32_t count_offset,  
uint32_t dispatch_offset,  
Flags flags = FlagNone,  
uint32_t max_size = Maxu32 )
```

dispatch inplace multiple indirect prefix scans

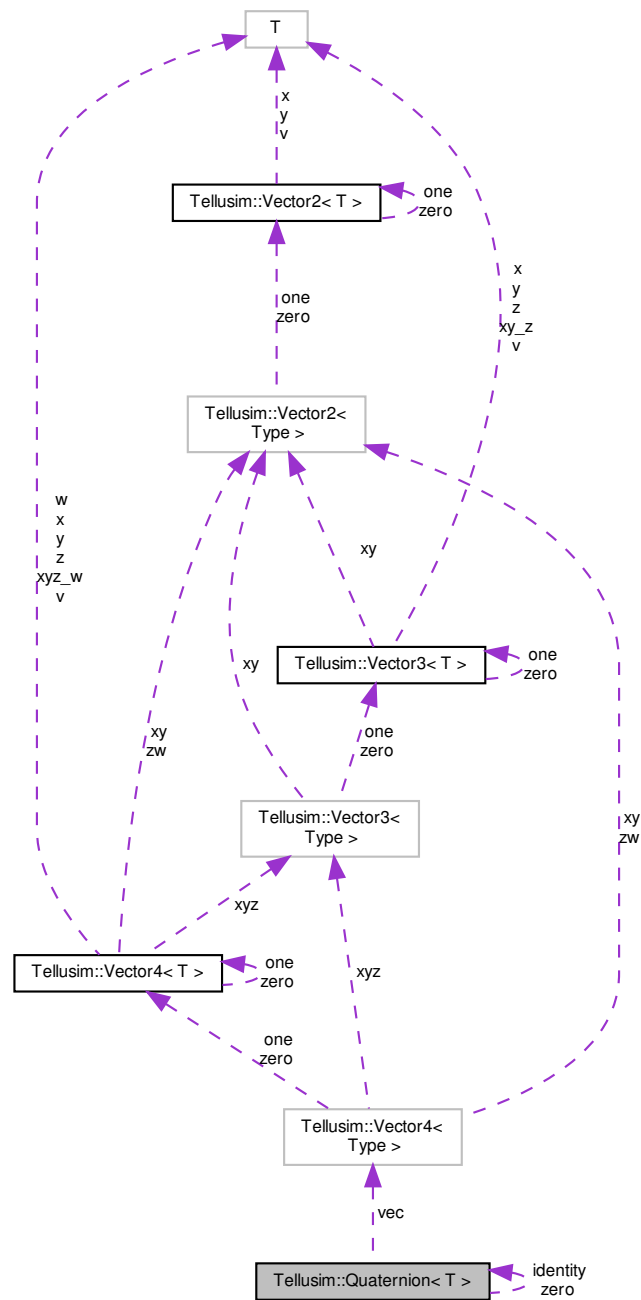
#### Parameters

|                        |                                           |
|------------------------|-------------------------------------------|
| <i>data</i>            | Buffer of uint32_t elements to scan.      |
| <i>count</i>           | Count indirect buffer.                    |
| <i>dispatch</i>        | Dispatch indirect buffer.                 |
| <i>count_offset</i>    | Count indirect buffer offset.             |
| <i>dispatch_offset</i> | Dispatch indirect buffer offset.          |
| <i>max_size</i>        | Maximum number of elements to scan.       |
| <i>repeat</i>          | Repeat dispatch with the last parameters. |

## 5.221 Tellusim::Quaternion< T > Struct Template Reference

```
#include <math/TellusimQuaternion.h>
```

Collaboration diagram for `Tellusim::Quaternion< T >`:



### Public Types

- enum { **Size** = 4 }
- using **Vector3** = `Tellusim::Vector3< Type >`
- using **Vector4** = `Tellusim::Vector4< Type >`
- using **Matrix4x3** = `Tellusim::Matrix4x3< Type >`
- using **Matrix4x4** = `Tellusim::Matrix4x4< Type >`



## Public Member Functions

- **Quaternion** (const [Quaternion](#) &q)
- **Quaternion** (Type x, Type y, Type z, Type w)
- **Quaternion** (const [Vector3](#) &axis, Type angle)
- **Quaternion** (const Type \*q)
- **Quaternion** (const [Matrix4x3](#) &m)
- **Quaternion** (const [Matrix4x4](#) &m)
- **Quaternion** (const [Vector4](#) &vector)
- template<class CType >  
**Quaternion** (const [Quaternion](#)< CType > &q)
- void **set** (Type X, Type Y, Type Z, Type W)  
*update quaternion data*
- void **set** (const Type \*1 q)
- void **get** (Type \*1 q) const
- void **setIdentity** ()  
*identity quaternion*
- bool **isIdentity** () const
- void **setRotateX** (Type angle)  
*rotation quaternion*
- void **setRotateY** (Type angle)
- void **setRotateZ** (Type angle)
- void **setRotateXYZ** (const [Vector3](#) &angles)
- void **setRotateZYX** (const [Vector3](#) &angles)
- void **setRotate** (const [Vector3](#) &axis, Type angle)
- Type **getRotateX** () const
- Type **getRotateY** () const
- Type **getRotateZ** () const
- [Vector3](#) **getRotateXYZ** () const
- [Vector3](#) **getRotateZYX** () const
- void **getRotate** ([Vector3](#) &axis, Type &angle) const
- void **setRotate** (Type x, Type y, Type z, Type angle)
- void **set** (const [Vector4](#) &1 row\_0, const [Vector4](#) &1 row\_1, const [Vector4](#) &1 row\_2)  
*quaternion to/from matrix*
- void **get** ([Vector4](#) &1 row\_0, [Vector4](#) &1 row\_1, [Vector4](#) &1 row\_2) const
- const Type & **operator[]** (uint32\_t index) const  
*quaternion data*
- Type & **operator[]** (uint32\_t index)

## Static Public Member Functions

- static [Quaternion](#) **rotateX** (Type angle)
- static [Quaternion](#) **rotateY** (Type angle)
- static [Quaternion](#) **rotateZ** (Type angle)
- static [Quaternion](#) **rotateXYZ** (const [Vector3](#) &angles)
- static [Quaternion](#) **rotateZYX** (const [Vector3](#) &angles)
- static [Quaternion](#) **rotate** (const [Vector3](#) &axis, Type angle)
- static [Quaternion](#) **rotateXYZ** (Type angle\_x, Type angle\_y, Type angle\_z)
- static [Quaternion](#) **rotateZYX** (Type angle\_x, Type angle\_y, Type angle\_z)
- static [Quaternion](#) **rotate** (Type axis\_x, Type axis\_y, Type axis\_z, Type angle)

## Public Attributes

- ```

union {
    struct {
        Type x
        Type y
        Type z
        Type w
    }
    Vector4 vec
    Type q [Size]
};

```

## Static Public Attributes

- static const **Quaternion** **zero**  
*default quaternions*
- static const **Quaternion** **identity**

## 5.221.1 Detailed Description

```

template<class T>
struct Tellusim::Quaternion< T >

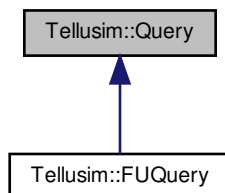
```

**Quaternion** class

## 5.222 Tellusim::Query Class Reference

```
#include <platform/TellusimQuery.h>
```

Inheritance diagram for Tellusim::Query:



## Classes

- struct **Statistics**  
*statistics query*

## Public Member Functions

- Platform [getPlatform](#) () const  
*query platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*query device index*
- void [clear](#) ()  
*clear query*
- bool [isCreated](#) () const  
*check query*
- bool **isAvailable** () const
- bool **isBegan** () const
- bool **isEnded** () const
- bool [create](#) (Type type)  
*create query*
- Type [getType](#) () const  
*query type*
- const char \* **getTypeName** () const
- size\_t **getTypeSize** () const
- bool **isTime** () const
- bool **isClock** () const
- bool **isSamples** () const
- bool **isSamples1** () const
- bool **isStatistics** () const
- bool **isTimeType** () const
- bool **isSamplesType** () const
- bool [get](#) (void \*dest, size\_t size, bool wait=true) const  
*get query data*
- uint64\_t **getTime** (bool wait=true, bool \*status=nullptr) const
- uint32\_t **getSamples** (bool wait=true, bool \*status=nullptr) const
- [Statistics](#) **getStatistics** (bool wait=true, bool \*status=nullptr) const

## Static Public Member Functions

- static const char \* **getTypeName** (Type type)

## 5.222.1 Detailed Description

[Query](#) class

## 5.223 Tellusim::RadixCompare&lt; Type &gt; Struct Template Reference

```
#include <core/TellusimSort.h>
```

## 5.223.1 Detailed Description

```
template<class Type>
struct Tellusim::RadixCompare< Type >
```

radixSort default compare

### 5.224 Tellusim::RadixCompare< float32\_t > Struct Template Reference

#### Public Types

- enum { **Bits** = 32 }
- using **Radix** = uint32\_t

#### Public Member Functions

- uint32\_t **operator()** (float32\_t value)

### 5.225 Tellusim::RadixCompare< int32\_t > Struct Template Reference

#### Public Types

- enum { **Bits** = 32 }
- using **Radix** = uint32\_t

#### Public Member Functions

- uint32\_t **operator()** (int32\_t value)

### 5.226 Tellusim::RadixCompare< uint32\_t > Struct Template Reference

#### Public Types

- enum { **Bits** = 32 }
- using **Radix** = uint32\_t

#### Public Member Functions

- uint32\_t **operator()** (uint32\_t value)

### 5.227 Tellusim::RadixMap< Key, Type, Size > Class Template Reference

```
#include <core/TellusimRadix.h>
```

#### Classes

- class [ConstIterator](#)  
*Constant iterator.*
- class [Iterator](#)  
*Iterator.*

## Public Member Functions

- [RadixMap](#) ()  
*constructors*
- **RadixMap** (const [RadixMap](#) &)=delete
- **RadixMap** ([RadixMap](#) &&map)
- void [clear](#) ()  
*clear map*
- void [swap](#) ([RadixMap](#) &map)  
*swap maps*
- void [move](#) ([RadixMap](#) &&map)  
*move map*
- [RadixMap](#) & **operator=** (const [RadixMap](#) &)=delete  
*assignment operators*
- [RadixMap](#) & **operator=** ([RadixMap](#) &&map)
- [Iterator](#) [append](#) (Key value)  
*append value*
- [Iterator](#) **append** (Key value, const Type &data)
- bool [remove](#) (uint32\_t value)  
*remove value*
- bool [empty](#) () const  
*map info*
- **operator bool** () const
- size\_t **memory** () const
- uint32\_t **size** () const
- [Iterator](#) [find](#) (Key value)  
*map data*
- [ConstIterator](#) **find** (Key value) const
- const Type & **operator[]** (Key value) const
- Type & **operator[]** (Key value)
- const Type & **get** (Key value) const
- Type & **get** (Key value)
- [Iterator](#) [begin](#) ()  
*map iterators*
- [Iterator](#) **back** ()
- [Iterator](#) **end** ()
- [ConstIterator](#) **begin** () const
- [ConstIterator](#) **back** () const
- [ConstIterator](#) **end** () const

## 5.227.1 Detailed Description

```
template<class Key, class Type, uint32_t Size = 32>
class Tellusim::RadixMap< Key, Type, Size >
```

[RadixMap](#) container

## 5.228 Tellusim::RadixSort Class Reference

```
#include <parallel/TellusimRadixSort.h>
```

## Classes

- struct [DispatchParameters](#)

## Public Types

- enum [Mode](#) {  
**ModeSingle** = 0,  
**ModeMultiple**,  
**NumModes** }  
*Sort modes.*
- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagSingle** = (1 << ModeSingle),  
**FlagMultiple** = (1 << ModeMultiple),  
**FlagIndirect** = (1 << (NumModes + 0)),  
**FlagOrder** = (1 << (NumModes + 1)),  
**FlagTracing** = (1 << (NumModes + 2)),  
**FlagScratch** = (1 << (NumModes + 3)),  
**FlagsAll** = (FlagSingle | FlagMultiple | FlagIndirect | FlagOrder) }  
*Sort flags.*

## Public Member Functions

- void [clear](#) ()  
*clear sort*
- bool [isCreated](#) ([Flags](#) flags) const  
*check sort*
- uint32\_t [getDataSize](#) () const  
*sort parameters*
- uint32\_t [getGroupSize](#) () const
- uint32\_t [getSortElements](#) () const
- uint32\_t [getUpdateElements](#) () const
- uint32\_t [getMaxElements](#) () const
- uint32\_t [getMaxRegions](#) () const
- [PrefixScan](#) [getPrefixScan](#) () const
- [Buffer](#) [getDataBuffer](#) () const
- bool [create](#) (const [Device](#) &device, [Mode](#) mode, [PrefixScan](#) &scan, uint32\_t size, uint32\_t groups=256, uint32\_t regions=1, [Async](#) \*async=NULLPTR)
- bool [create](#) (const [Device](#) &device, [Flags](#) flags, [PrefixScan](#) &scan, uint32\_t size, uint32\_t groups=256, uint32\_t regions=1, [Async](#) \*async=NULLPTR)
- bool [dispatch](#) ([Compute](#) &compute, [Buffer](#) &data, uint32\_t keys\_offset, uint32\_t data\_offset, uint32\_t size, [Flags](#) flags=FlagNone, uint32\_t bits=32)
- bool [dispatch](#) ([Compute](#) &compute, [Buffer](#) &data, uint32\_t count, const uint32\_t \*keys\_offsets, const uint32\_t \*data\_offsets, const uint32\_t \*sizes, [Flags](#) flags=FlagNone, uint32\_t bits=32)
- bool [dispatchIndirect](#) ([Compute](#) &compute, [Buffer](#) &data, [Buffer](#) &dispatch, uint32\_t offset, [Flags](#) flags=FlagNone, uint32\_t bits=32, uint32\_t max\_size=Maxu32)
- bool [dispatchIndirect](#) ([Compute](#) &compute, [Buffer](#) &data, uint32\_t count, [Buffer](#) &dispatch, uint32\_t offset, [Flags](#) flags=FlagNone, uint32\_t bits=32, uint32\_t max\_size=Maxu32)
- bool [dispatchIndirect](#) ([Compute](#) &compute, [Buffer](#) &data, [Buffer](#) &count, [Buffer](#) &dispatch, uint32\_t count\_offset, uint32\_t dispatch\_offset, [Flags](#) flags=FlagNone, uint32\_t bits=32, uint32\_t max\_size=Maxu32)

## 5.228.1 Detailed Description

[RadixSort](#) class

## 5.228.2 Member Function Documentation

## 5.228.2.1 create()

```
bool Tellusim::RadixSort::create (
    const Device & device,
    Mode mode,
    PrefixScan & scan,
    uint32_t size,
    uint32_t groups = 256,
    uint32_t regions = 1,
    Async * async = nullptr )
```

create radix sort

## Parameters

<i>scan</i>	Prefix scan.
<i>size</i>	Radix sort data size.
<i>groups</i>	Radix sort group size.
<i>regions</i>	Maximum number of multiple regions.

## 5.228.2.2 dispatch() [1/2]

```
bool Tellusim::RadixSort::dispatch (
    Compute & compute,
    Buffer & data,
    uint32_t keys_offset,
    uint32_t data_offset,
    uint32_t size,
    Flags flags = FlagNone,
    uint32_t bits = 32 )
```

dispatch inplace single radix sort

## Parameters

<i>data</i>	<a href="#">Buffer</a> of uint32_t data elements to sort.
<i>keys_offset</i>	Keys elements offset index (4 aligned).
<i>data_offset</i>	Data elements offset index (4 aligned).
<i>size</i>	Number of uint32_t elements to sort.
<i>bits</i>	Number of key bits to sort.

5.228.2.3 `dispatch()` [2/2]

```
bool Tellusim::RadixSort::dispatch (
    Compute & compute,
    Buffer & data,
    uint32_t count,
    const uint32_t * keys_offsets,
    const uint32_t * data_offsets,
    const uint32_t * sizes,
    Flags flags = FlagNone,
    uint32_t bits = 32 )
```

dispatch inplace multiple radix sorts

## Parameters

<i>data</i>	Buffer of uint32_t data elements to sort.
<i>count</i>	Number of regions to sort.
<i>keys_offsets</i>	Keys elements offset index (4 aligned).
<i>data_offsets</i>	Data elements offset index (4 aligned).
<i>sizes</i>	Number of uint32_t elements to sort.
<i>bits</i>	Number of key bits to sort.

5.228.2.4 `dispatchIndirect()` [1/3]

```
bool Tellusim::RadixSort::dispatchIndirect (
    Compute & compute,
    Buffer & data,
    Buffer & dispatch,
    uint32_t offset,
    Flags flags = FlagNone,
    uint32_t bits = 32,
    uint32_t max_size = Maxu32 )
```

dispatch inplace single indirect radix sort

## Parameters

<i>data</i>	Buffer of uint32_t data elements to sort.
<i>dispatch</i>	Dispatch indirect buffer.
<i>offset</i>	Dispatch indirect buffer offset.
<i>bits</i>	Number of key bits to sort.
<i>max_size</i>	Maximum number of elements to sort.

5.228.2.5 `dispatchIndirect()` [2/3]

```
bool Tellusim::RadixSort::dispatchIndirect (
```



```

    Compute & compute,
    Buffer & data,
    uint32_t count,
    Buffer & dispatch,
    uint32_t offset,
    Flags flags = FlagNone,
    uint32_t bits = 32,
    uint32_t max_size = Maxu32 )

```

dispatch inplace multiple indirect radix sorts

#### Parameters

<i>data</i>	Buffer of uint32_t data elements to sort.
<i>count</i>	Number of regions to sort.
<i>dispatch</i>	Dispatch indirect buffer.
<i>offset</i>	Dispatch indirect buffer offset.
<i>bits</i>	Number of key bits to sort.
<i>max_size</i>	Maximum number of elements to sort.

#### 5.228.2.6 dispatchIndirect() [3/3]

```

bool Tellusim::RadixSort::dispatchIndirect (
    Compute & compute,
    Buffer & data,
    Buffer & count,
    Buffer & dispatch,
    uint32_t count_offset,
    uint32_t dispatch_offset,
    Flags flags = FlagNone,
    uint32_t bits = 32,
    uint32_t max_size = Maxu32 )

```

dispatch inplace multiple indirect radix sorts

#### Parameters

<i>data</i>	Buffer of uint32_t data elements to sort.
<i>count</i>	Count indirect buffer.
<i>dispatch</i>	Dispatch indirect buffer.
<i>count_offset</i>	Count indirect buffer offset.
<i>dispatch_offset</i>	Dispatch indirect buffer offset.
<i>bits</i>	Number of key bits to sort.
<i>max_size</i>	Maximum number of elements to sort.

## 5.229 Tellusim::Random< Integer, Float > Struct Template Reference

```
#include <math/TellusimRandom.h>
```

## Public Types

- enum { **MaxValue** = 0x0ffffff }

## Public Member Functions

- [Random](#) ()  
*constructor*
- **Random** (const Integer &s)
- void [init](#) (const Integer &s)  
*initialize random*
- Integer [geti32](#) (int32\_t mask=MaxValue)  
*returns an integer number*
- Integer [geti32](#) (const Integer &min, const Integer &max)  
*returns an integer in [min-max] range*
- Float [getf32](#) ()  
*returns a floating-point number in [0-1] range*
- Float [getf32](#) (const Float &min, const Float &max)  
*returns a floating-point number in [min-max] range*

## Public Attributes

- Integer **seed\_0**
- Integer **seed\_1**

## 5.229.1 Detailed Description

```
template<class Integer = int32_t, class Float = float32_t>
struct Tellusim::Random< Integer, Float >
```

Linear congruential random generator

## 5.230 Tellusim::Rect Struct Reference

```
#include <interface/TellusimTypes.h>
```

## Public Member Functions

- **Rect** (float32\_t value)
- **Rect** (float32\_t horizontal, float32\_t vertical)
- **Rect** (float32\_t left, float32\_t right, float32\_t bottom, float32\_t top)
- bool **isValid** () const  
*check rectangle*
- **operator bool** () const
- **Rect & expand** (float32\_t x, float32\_t y)  
*expand by point*
- **Rect & expand** (const **Rect** &rect)  
*expand by rectangle*
- **Rect & shrink** (const **Rect** &rect)  
*shrink by rectangle*
- bool **inside** (float32\_t x, float32\_t y) const  
*inside point*
- bool **inside** (const **Vector2f** &v) const
- bool **inside** (const **Rect** &rect) const
- **Vector2f** **getSize** () const  
*rectangle size*
- float32\_t **getWidth** () const
- float32\_t **getHeight** () const
- **Vector2f** **getCenter** () const  
*center of rectangle*
- float32\_t **getCenterX** () const
- float32\_t **getCenterY** () const
- **Rect & operator+=** (const **Vector2f** &v)  
*rectangle to vector operators*
- **Rect & operator-=** (const **Vector2f** &v)
- **Rect & operator+=** (const **Rect** &r)  
*rectangle to rectangle operators*
- **Rect & operator-=** (const **Rect** &r)

## Public Attributes

```

•
union {
    struct {
        float32_t left
        float32_t right
        float32_t bottom
        float32_t top
    }
    float32_t rect [4]
};

```

## 5.230.1 Detailed Description

**Rect**

### 5.231 Tellusim::Region Struct Reference

```
#include <TellusimTypes.h>
```

#### Public Member Functions

- **Region** (uint32\_t width, uint32\_t height)
- **Region** (uint32\_t width, uint32\_t height, uint32\_t depth)
- **Region** (uint32\_t x, uint32\_t y, uint32\_t width, uint32\_t height)
- **Region** (uint32\_t x, uint32\_t y, uint32\_t z, uint32\_t width, uint32\_t height, uint32\_t depth)
- **Region** (const [Origin](#) &origin, const [Size](#) &size)
- **Region** (const [Size](#) &size)
- [Origin](#) **getOrigin** () const  
*region parameters*
- [Size](#) **getSize** () const

#### Public Attributes

- uint32\_t **x** = 0
- uint32\_t **y** = 0
- uint32\_t **z** = 0
- uint32\_t **width** = 0
- uint32\_t **height** = 0
- uint32\_t **depth** = 0

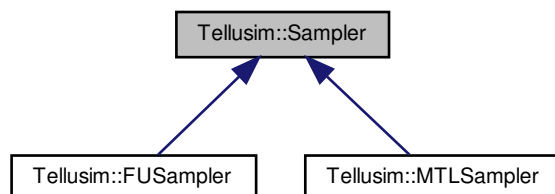
#### 5.231.1 Detailed Description

[Region](#)

### 5.232 Tellusim::Sampler Class Reference

```
#include <platform/TellusimSampler.h>
```

Inheritance diagram for Tellusim::Sampler:



## Public Types

- enum [Filter](#) {  
**FilterPoint** = 0,  
**FilterLinear**,  
**FilterBipoint**,  
**FilterBilinear**,  
**FilterTrilinear**,  
**NumFilters** }  
*Filter types.*
- enum {  
**MinAnisotropy** = 1,  
**MaxAnisotropy** = 16 }  
*Anisotropy range.*
- enum [WrapMode](#) {  
**WrapModeClamp** = 0,  
**WrapModeRepeat**,  
**WrapModeMirror**,  
**WrapModeBorder**,  
**NumWrapModes** }  
*Wrap modes.*
- enum [CompareFunc](#) {  
**CompareFuncNone** = 0,  
**CompareFuncEqual**,  
**CompareFuncLess**,  
**CompareFuncGreater**,  
**CompareFuncNotEqual**,  
**CompareFuncLessEqual**,  
**CompareFuncGreaterEqual**,  
**NumCompareFunctions** }  
*Compare functions.*
- enum [ReductionMode](#) {  
**ReductionModeAverage** = 0,  
**ReductionModeMin**,  
**ReductionModeMax**,  
**NumReductionModes** }  
*Reduction modes.*

## Public Member Functions

- Platform [getPlatform](#) () const  
*sampler platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*sampler device index*
- void [clear](#) ()  
*clear sampler*
- bool [isCreated](#) () const  
*check sampler*
- bool [create](#) ()  
*create sampler*
- void [setParameters](#) (const [Sampler](#) &sampler)  
*sampler parameters*
- void [setFilter](#) ([Filter](#) filter)

- filter type*
  - [Filter](#) **getFilter** () const
  - bool **isPointFilter** () const
  - void **setAnisotropy** (uint32\_t anisotropy)
- anisotropy level*
  - uint32\_t **getAnisotropy** () const
  - bool **hasAnisotropy** () const
  - void **setWrapMode** ([WrapMode](#) mode)
- wrapping mode*
  - void **setWrapMode** ([WrapMode](#) mode\_s, [WrapMode](#) mode\_t, [WrapMode](#) mode\_r)
  - void **setWrapModeS** ([WrapMode](#) mode)
  - [WrapMode](#) **getWrapModeS** () const
  - void **setWrapModeT** ([WrapMode](#) mode)
  - [WrapMode](#) **getWrapModeT** () const
  - void **setWrapModeR** ([WrapMode](#) mode)
  - [WrapMode](#) **getWrapModeR** () const
  - void **setLod** (float32\_t min, float32\_t max, float32\_t bias)
- level of detail*
  - void **setLodMin** (float32\_t min)
  - float32\_t **getLodMin** () const
  - void **setLodMax** (float32\_t max)
  - float32\_t **getLodMax** () const
  - void **setLodBias** (float32\_t bias)
  - float32\_t **getLodBias** () const
  - void **setBorderColor** (const [Color](#) &color)
- border color*
  - void **setBorderColor** (float32\_t r, float32\_t g, float32\_t b, float32\_t a)
  - const [Color](#) & **getBorderColor** () const
  - void **setCompareFunc** ([CompareFunc](#) func)
- compare func*
  - [CompareFunc](#) **getCompareFunc** () const
  - void **setReductionMode** ([ReductionMode](#) mode)
- reduction mode*
  - [ReductionMode](#) **getReductionMode** () const

#### 5.232.1 Detailed Description

[Sampler](#) class

#### 5.233 Tellusim::Scissor Struct Reference

```
#include <TellusimTypes.h>
```

##### Public Member Functions

- **Scissor** (int32\_t width, int32\_t height)
- **Scissor** (int32\_t x, int32\_t y, int32\_t width, int32\_t height)

## Public Attributes

- `int32_t x = 0`
- `int32_t y = 0`
- `int32_t width = Maxi16`
- `int32_t height = Maxi16`

## 5.233.1 Detailed Description

[Scissor](#)

## 5.234 Tellusim::SeparableFilter Class Reference

```
#include <graphics/TellusimSeparableFilter.h>
```

## Public Types

- enum [Mode](#) {  
**ModeHorizontal** = 0,  
**ModeVertical**,  
**NumModes** }  
*Filter modes.*
- enum [Flags](#) {  
**FlagNone** = 0,  
**FlagRepeat** = (1 << 0),  
**FlagZero** = (1 << 1),  
**DefaultFlags** = FlagNone }  
*Filter flags.*

## Public Member Functions

- void [clear](#) ()  
*clear filter*
- bool [isCreated](#) (Format format, uint32\_t size) const  
*check filter*
- void [setInputSource](#) ([Mode](#) mode, const char \*src)
- [String](#) [getInputSource](#) ([Mode](#) mode) const
- void [setOutputSource](#) ([Mode](#) mode, const char \*src)
- [String](#) [getOutputSource](#) ([Mode](#) mode) const
- bool [create](#) (const [Device](#) &device, Format format, uint32\_t size, [Flags](#) flags=DefaultFlags)
- void [setWeights](#) ([Mode](#) mode, const Array< [Vector4f](#) > &weights, bool normalize=false)  
*filter weights*
- void [setWeights](#) ([Mode](#) mode, const Array< float32\_t > &weights, bool normalize=false)
- void [setGaussianWeights](#) (uint32\_t size, const [Vector4f](#) &sigma)  
*Gaussian filter weights.*
- void [setGaussianWeights](#) (uint32\_t size, float32\_t sigma)
- void [setSobelXWeights](#) (uint32\_t size)  
*Sobel filter weights.*
- void [setSobelYWeights](#) (uint32\_t size)
- void [setBoxWeights](#) (uint32\_t size)  
*box filter weights*
- bool [dispatch](#) ([Compute](#) &compute, [Mode](#) mode, uint32\_t size, [Texture](#) &dest, [Texture](#) &src, const [Slice](#) &dest\_slice, const [Slice](#) &src\_slice, const [Vector4f](#) &values=[Vector4f::zero](#)) const
- bool [dispatch](#) ([Compute](#) &compute, [Mode](#) mode, uint32\_t size, [Texture](#) &dest, [Texture](#) &src, const [Slice](#) &src\_slice, const [Vector4f](#) &values=[Vector4f::zero](#)) const
- bool [dispatch](#) ([Compute](#) &compute, [Mode](#) mode, uint32\_t size, [Texture](#) &dest, [Texture](#) &src, const [Vector4f](#) &values=[Vector4f::zero](#)) const

### 5.234.1 Detailed Description

[SeparableFilter](#) class

### 5.234.2 Member Function Documentation

#### 5.234.2.1 setInputSource()

```
void Tellusim::SeparableFilter::setInputSource (
    Mode mode,
    const char * src )
```

input shader source

#### Parameters

<i>src</i>	<a href="#">Shader</a> source.
------------	--------------------------------

#### 5.234.2.2 setOutputSource()

```
void Tellusim::SeparableFilter::setOutputSource (
    Mode mode,
    const char * src )
```

output shader source

#### Parameters

<i>src</i>	<a href="#">Shader</a> source.
------------	--------------------------------

#### 5.234.2.3 create()

```
bool Tellusim::SeparableFilter::create (
    const Device & device,
    Format format,
    uint32_t size,
    Flags flags = DefaultFlags )
```

create filter

#### Parameters

<i>format</i>	<a href="#">Texture</a> format.
<i>size</i>	Filter size in pixels, the actual filter size is (size * 2 + 1).



## 5.234.2.4 dispatch()

```
bool Tellusim::SeparableFilter::dispatch (
    Compute & compute,
    Mode mode,
    uint32_t size,
    Texture & dest,
    Texture & src,
    const Slice & dest_slice,
    const Slice & src_slice,
    const Vector4f & values = Vector4f::zero ) const
```

dispatch separable filter

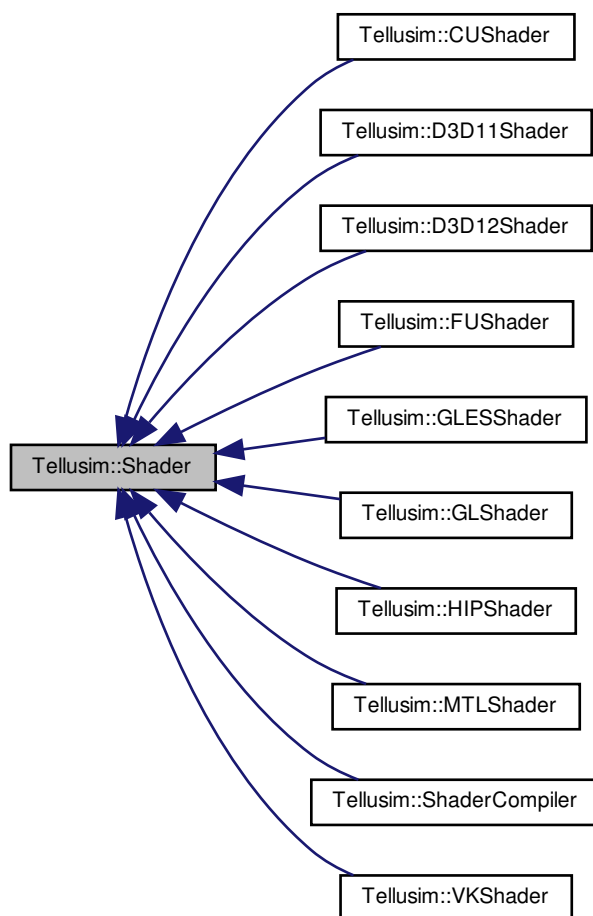
## Parameters

<i>dest</i>	Destination texture.
<i>src</i>	Source texture.
<i>dest_slice</i>	Destination texture slice.
<i>src_slice</i>	Source texture slice.
<i>values</i>	Filter parameters available for the source blocks.

## 5.235 Tellusim::Shader Class Reference

```
#include <platform/TellusimShader.h>
```

Inheritance diagram for Tellusim::Shader:



## Public Types

- enum [Mask](#) {
  - MaskNone** = 0,
  - MaskVertex** = (1 << TypeVertex),
  - MaskControl** = (1 << TypeControl),
  - MaskEvaluate** = (1 << TypeEvaluate),
  - MaskGeometry** = (1 << TypeGeometry),
  - MaskFragment** = (1 << TypeFragment),
  - MaskCompute** = (1 << TypeCompute),
  - MaskTask** = (1 << TypeTask),
  - MaskMesh** = (1 << TypeMesh),
  - MaskRayGen** = (1 << TypeRayGen),
  - MaskRayMiss** = (1 << TypeRayMiss),
  - MaskClosest** = (1 << TypeClosest),
  - MaskFirstHit** = (1 << TypeFirstHit),
  - MaskIntersection** = (1 << TypeIntersection),
  - MaskCallable** = (1 << TypeCallable),

```

MaskVertexFragment = (MaskVertex | MaskFragment),
MaskGraphics = (MaskVertex | MaskControl | MaskEvaluate | MaskGeometry | MaskFragment),
MaskTracing = (MaskRayGen | MaskRayMiss | MaskClosest | MaskFirstHit | MaskIntersection | MaskCallable),
MaskAll = (MaskGraphics | MaskCompute | MaskTask | MaskMesh | MaskTracing) }

```

*Shader masks.*

## Public Member Functions

- Platform [getPlatform](#) () const  
*shader platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*shader device index*
- void [clear](#) ()  
*clear shader*
- bool [isCreated](#) () const  
*check shader*
- bool [saveState](#) ([Stream](#) &stream) const  
*shader parameters*
- Type [getType](#) () const  
*shader type*
- const char \* **getTypeName** () const
- bool **isVertex** () const
- bool **isControl** () const
- bool **isEvaluate** () const
- bool **isGeometry** () const
- bool **isFragment** () const
- bool **isCompute** () const
- bool **isTask** () const
- bool **isMesh** () const
- bool **isRayGen** () const
- bool **isRayMiss** () const
- bool **isClosest** () const
- bool **isFirstHit** () const
- bool **isIntersection** () const
- bool **isCallable** () const
- bool **isGraphicsType** () const
- bool **isTessellationType** () const
- bool **isTracingType** () const
- bool **isMeshType** () const
- [String](#) [getName](#) () const  
*shader name*
- [String](#) [getMacros](#) () const
- void [setSamplerOffset](#) (int32\_t offset)  
*shader samplers*
- int32\_t [getSamplerOffset](#) () const
- void [setTextureOffset](#) (int32\_t offset)  
*shader textures*
- int32\_t [getTextureOffset](#) () const
- void [setSurfaceOffset](#) (int32\_t offset)  
*shader surfaces*

- int32\_t **getSurfaceOffset** () const
- void **setUniformOffset** (int32\_t offset)
  - shader uniforms*
- int32\_t **getUniformOffset** () const
- void **setStorageOffset** (int32\_t offset)
  - shader storages*
- int32\_t **getStorageOffset** () const
- void **setTracingOffset** (int32\_t offset)
  - shader tracings*
- int32\_t **getTracingOffset** () const
- void **setTexelOffset** (int32\_t offset)
  - shader texels*
- int32\_t **getTexelOffset** () const
- void **setTableOffset** (int32\_t offset)
  - shader tables*
- int32\_t **getTableOffset** () const
- void **setPatchSize** (uint32\_t size)
  - shader patch size*
- uint32\_t **getPatchSize** () const
- void **setInputSize** (uint32\_t size)
  - shader input size*
- uint32\_t **getInputSize** () const
- void **setOutputSize** (uint32\_t size)
  - shader output size*
- uint32\_t **getOutputSize** () const
- bool **load** (Type type, const char \*name, const char \*format,...) 1(4)
  - create native shader*
- bool bool **create** (Type type, const char \*src, const char \*format,...) 1(4)
- bool bool bool **load** (Type type, const char \*name, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool **create** (Type type, const char \*src, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool **loadGLSL** (Type type, const char \*name, const char \*format,...) 1(4)
  - create GLSL shader*
- bool bool **createGLSL** (Type type, const char \*src, const char \*format,...) 1(4)
- bool bool bool **loadGLSL** (Type type, const char \*name, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool **createGLSL** (Type type, const char \*src, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool **loadSPIRV** (Type type, const char \*name)
  - create SPIR-V shader*
- bool **createSPIRV** (Type type, const Array< uint32\_t > &data)

#### Static Public Member Functions

- static const char \* **getTypeName** (Type type)
- static bool **hasCache** ()
  - global shader cache*
- static bool **setCache** (const char \*name)
- static bool **loadCache** (const [String](#) &hash, [Stream](#) &stream)
- static bool **saveCache** (const [String](#) &hash, [Stream](#) &stream)
- static void **clearCache** ()

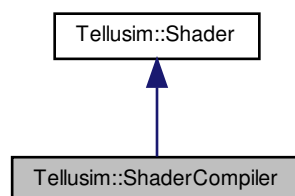
- static bool [isMacro](#) (const char \*name)  
*global macro definitions*
- static bool **setMacro** (const char \*name, int32\_t value)
- static bool **setMacro** (const char \*name, uint32\_t value)
- static bool **setMacro** (const char \*name, const char \*value=NULLPTR)
- static bool **setMacros** (const char \*macros)
- static bool **removeMacro** (const char \*name)
- static void **clearMacros** ()
- static bool [isInclude](#) (const char \*name)  
*global include definitions*
- static bool **setInclude** (const char \*name, const [String](#) &src)
- static bool **removeInclude** (const char \*name)
- static void **clearIncludes** ()
- static [String](#) [preprocessor](#) (const char \*src, const char \*format,...) 1(2)  
*global macro preprocessor*
- static [String](#) static [String](#) **preprocessor** (const char \*src, const [String](#) &macros=[String::null](#), const char \*\*includes=NULLPTR, uint32\_t size=0)

#### 5.235.1 Detailed Description

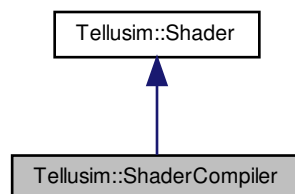
[Shader](#) class

### 5.236 Tellusim::ShaderCompiler Class Reference

Inheritance diagram for Tellusim::ShaderCompiler:



Collaboration diagram for Tellusim::ShaderCompiler:



## Public Types

- enum **Flags** {  
**FlagNone** = 0,  
**FlagMSLIndirect** = (1 << 0) }

## Public Member Functions

- void [setFlags](#) (Flags flags)  
*shader flags*
- Flags **getFlags** () const
- bool [getBinary](#) ([Stream](#) &stream, Platform platform=PlatformUnknown) const  
*shader binary*
- [String](#) [getSource](#) (Platform platform=PlatformUnknown) const  
*shader source*

## Additional Inherited Members

## 5.237 Tellusim::Size Struct Reference

```
#include <TellusimTypes.h>
```

## Public Member Functions

- **Size** (uint32\_t size)
- **Size** (uint32\_t width, uint32\_t height)
- **Size** (uint32\_t width, uint32\_t height, uint32\_t depth)

## Public Attributes

- uint32\_t **width** = 0
- uint32\_t **height** = 0
- uint32\_t **depth** = 0

## 5.237.1 Detailed Description

[Size](#)

## 5.238 Tellusim::Slice Struct Reference

## Public Member Functions

- **Slice** (const [Face](#) &f)
- **Slice** (const [Layer](#) &l)
- **Slice** (const [Mipmap](#) &m)
- **Slice** (const [Layer](#) &l, const [Face](#) &f)
- **Slice** (const [Face](#) &f, const [Mipmap](#) &m)
- **Slice** (const [Layer](#) &l, const [Mipmap](#) &m)
- **Slice** (const [Layer](#) &l, const [Face](#) &f, const [Mipmap](#) &m)
- bool [hasFaces](#) () const
- slice parameters*
- bool [hasLayers](#) () const
- bool [hasMipmaps](#) () const
- [Face](#) [getFace](#) () const
- [Layer](#) [getLayer](#) () const
- [Mipmap](#) [getMipmap](#) () const
- [Slice](#) [setFace](#) (uint32\_t base, uint32\_t size=1) const
- set slice parameters*
- [Slice](#) [setLayer](#) (uint32\_t base, uint32\_t size=1) const
- [Slice](#) [setMipmap](#) (uint32\_t base, uint32\_t size=1) const
- [Slice](#) [setSize](#) (const [Slice](#) &s) const
- replace slice sizes*
- [Slice](#) [addBase](#) (const [Slice](#) &s) const
- increment slice bases*

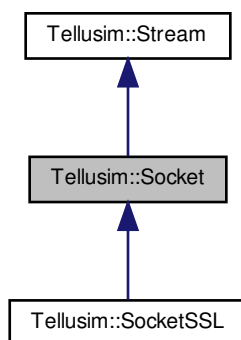
## Public Attributes

- uint32\_t **face** = 0
- uint32\_t **faces** = 1
- uint32\_t **layer** = 0
- uint32\_t **layers** = 1
- uint32\_t **mipmap** = 0
- uint32\_t **mipmaps** = 1

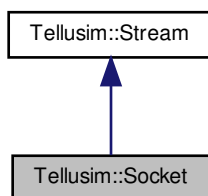
## 5.239 Tellusim::Socket Class Reference

```
#include <core/TellusimSocket.h>
```

Inheritance diagram for Tellusim::Socket:



Collaboration diagram for Tellusim::Socket:



#### Public Member Functions

- **Socket** (Type type=TypeStream)
- bool **open** (uint16\_t port, uint16\_t num=32)  
*open/close socket*
- bool **open** (const char \*name, uint16\_t port)
- bool **open** (const **String** &name, uint16\_t port)
- void **close** ()
- bool **connect** (uint32\_t sec, uint32\_t usec=0)  
*stream socket*
- virtual bool **accept** (**Socket** &socket)
- bool **select** (uint32\_t sec, uint32\_t usec=0)  
*socket operations*
- bool **setTimeout** (uint32\_t sec)  
*socket timeout*
- uint32\_t **getTimeout** () const
- bool **setBlock** (bool block)



*socket blocking*

- bool **getBlock** () const
- bool **setDelay** (bool delay)

*socket delay*

- bool **getDelay** () const
- void **setName** (const char \*name)

*socket parameters*

- void **setName** (const [String](#) &name)
- uint16\_t **getPort** () const
- int32\_t **getFD** () const
- Type **getType** () const

### Static Public Member Functions

- static [String](#) **getAddress** (const char \*delimiter=nullptr)

*socket utils*

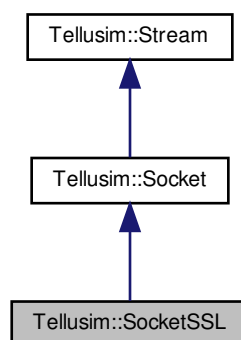
### 5.239.1 Detailed Description

[Socket](#) class

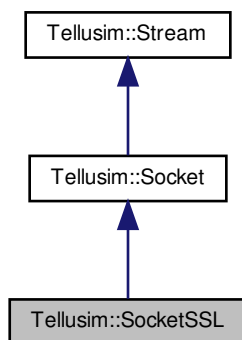
## 5.240 Tellusim::SocketSSL Class Reference

```
#include <core/TellusimSocket.h>
```

Inheritance diagram for Tellusim::SocketSSL:



Collaboration diagram for Tellusim::SocketSSL:



#### Public Member Functions

- bool [handshake](#) (const char \*name=nullptr)  
*stream socket*
- virtual bool **accept** ([SocketSSL](#) &socket)
- virtual bool **accept** ([Socket](#) &socket)
- bool [load](#) ([Stream](#) &stream)  
*socket certificate*
- bool **load** (const char \*name)
- bool **load** (const [String](#) &name)
- bool [isConnected](#) () const  
*socket status*

#### Additional Inherited Members

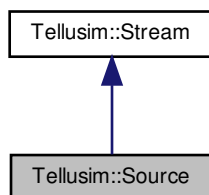
##### 5.240.1 Detailed Description

[SocketSSL](#) class

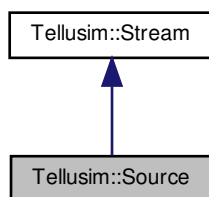
##### 5.241 Tellusim::Source Class Reference

```
#include <core/TellusimSource.h>
```

Inheritance diagram for Tellusim::Source:



Collaboration diagram for Tellusim::Source:



### Public Types

- using **IsCallback** = bool(const char \*name, void \*data)  
*source callback*
- using **OpenCallback** = **Stream**(const char \*name, void \*data)

### Public Member Functions

- **Source** (const uint8\_t \*data, size\_t size, const char \*name=nullptr)
- bool **open** (const char \*name, bool callback=true, bool write=false)  
*open/close source*
- bool **open** (const **String** &name, bool callback=true, bool write=false)
- void **close** ()
- void **setName** (const char \*name, size\_t offset, size\_t size)  
*source name*
- void **setName** (const **String** &name, size\_t offset, size\_t size)
- void **setData** (const uint8\_t \*data, size\_t size, const char \*name=nullptr)  
*source data*

### Static Public Member Functions

- static bool [isSource](#) (const char \*name)  
*source utils*
- static bool **isSource** (const [String](#) &name)
- static uint64\_t **getTime** (const char \*name)
- static size\_t **getSize** (const char \*name)
- static void **setCallback** (OpenCallback \*open\_func, void \*data=nullptr)
- static void **setCallback** (OpenCallback \*open\_func, [IsCallback](#) \*is\_func, void \*data=nullptr)
- static OpenCallback \* **getOpenCallback** ()
- static [IsCallback](#) \* **getIsCallback** ()
- static void \* **getCallbackData** ()

#### 5.241.1 Detailed Description

[Source](#) class

### 5.242 Tellusim::SpatialGrid Class Reference

```
#include <parallel/TellusimSpatialGrid.h>
```

#### Classes

- struct [DispatchParameters](#)

#### Public Member Functions

- void [clear](#) ()  
*clear grid*
- bool [isCreated](#) () const  
*check grid*
- uint32\_t [getGroupSize](#) () const  
*grid parameters*
- [RadixSort](#) **getRadixSort** () const
- bool [create](#) (const [Device](#) &device, [RadixSort](#) &sort, uint32\_t groups=256)  
*create spatial grid*
- bool [dispatch](#) ([Compute](#) &compute, [Buffer](#) &data, uint32\_t offset, uint32\_t size, uint32\_t bits=32)
- bool [dispatchIndirect](#) ([Compute](#) &compute, [Buffer](#) &data, [Buffer](#) &dispatch, uint32\_t offset, uint32\_t max\_↔ size=Maxu32)

#### 5.242.1 Detailed Description

[SpatialGrid](#) class

#### 5.242.2 Member Function Documentation

## 5.242.2.1 dispatch()

```
bool Tellusim::SpatialGrid::dispatch (
    Compute & compute,
    Buffer & data,
    uint32_t offset,
    uint32_t size,
    uint32_t bits = 32 )
```

dispatch spatial grid generation

## Parameters

<i>data</i>	<a href="#">Spatial</a> grid of uint32_t elements.
<i>offset</i>	<a href="#">Spatial</a> grid offset index.
<i>size</i>	Number of spatial elements.
<i>bits</i>	Number of hash bits to sort.

## 5.242.2.2 dispatchIndirect()

```
bool Tellusim::SpatialGrid::dispatchIndirect (
    Compute & compute,
    Buffer & data,
    Buffer & dispatch,
    uint32_t offset,
    uint32_t max_size = Maxu32 )
```

dispatch spatial tree generation

## Parameters

<i>data</i>	<a href="#">Spatial</a> grid of uint32_t elements.
<i>dispatch</i>	Dispatch indirect buffer.
<i>offset</i>	Dispatch indirect buffer offset.
<i>max_size</i>	Maximum number of spatial elements.

## 5.243 Tellusim::SpatialTree Class Reference

```
#include <parallel/TellusimSpatialTree.h>
```

## Classes

- struct [DispatchParameters](#)
- struct [Node](#)

## Public Types

- enum `Mode` {  
**ModeSingle** = 0,  
**ModeMultiple**,  
**NumModes** }  
*Tree modes.*
- enum `Hash` {  
**HashXYZ10**,  
**HashXYZ9**,  
**HashXYZ8**,  
**HashXY15**,  
**HashXY14**,  
**HashXY12**,  
**HashXY10** }  
*Hash modes.*
- enum `Flags` {  
**FlagNone** = 0,  
**FlagSingle** = (1 << ModeSingle),  
**FlagMultiple** = (1 << ModeMultiple),  
**FlagUpdate** = (1 << (NumModes + 0)),  
**FlagOptimize** = (1 << (NumModes + 1)),  
**FlagTracing** = (1 << (NumModes + 2)),  
**FlagScratch** = (1 << (NumModes + 3)),  
**FlagAtomic** = (1 << (NumModes + 4)),  
**FlagSingleUpdate** = (FlagSingle | FlagUpdate),  
**FlagMultipleUpdate** = (FlagMultiple | FlagUpdate),  
**FlagSingleOptimize** = (FlagSingle | FlagOptimize),  
**FlagMultipleOptimize** = (FlagMultiple | FlagOptimize),  
**FlagsAll** = (FlagSingle | FlagMultiple),  
**FlagsAllOptimize** = (FlagsAll | FlagOptimize) }  
*Tree flags.*

## Public Member Functions

- void `clear` ()  
*clear tree*
- bool `isCreated` (`Flags` flags) const  
*check tree*
- uint32\_t `getGroupSize` () const  
*tree parameters*
- uint32\_t `getBoundsNodes` () const
- uint32\_t `getMaxNodes` () const
- uint32\_t `getMaxRegions` () const
- `RadixSort` `getRadixSort` () const
- bool `create` (const `Device` &device, `Mode` mode, `RadixSort` &sort, uint32\_t size, uint32\_t groups=256, uint32\_t regions=1, `Async` \*async=NULLPTR)
- bool `create` (const `Device` &device, `Flags` flags, `RadixSort` &sort, uint32\_t size, uint32\_t groups=256, uint32\_t regions=1, `Async` \*async=NULLPTR)
- bool `dispatch` (`Compute` &compute, `Hash` hash, `Buffer` &nodes, uint32\_t offset, uint32\_t size, `Flags` flags=FlagNone)
- bool `dispatch` (`Compute` &compute, `Hash` hash, `Buffer` &nodes, uint32\_t count, const uint32\_t \*offsets, const uint32\_t \*sizes, `Flags` flags=FlagNone)
- bool `dispatchIndirect` (`Compute` &compute, `Hash` hash, `Buffer` &nodes, `Buffer` &dispatch, uint32\_t offset, uint32\_t max\_size=Maxu32)
- bool `dispatchIndirect` (`Compute` &compute, `Hash` hash, `Buffer` &nodes, uint32\_t count, `Buffer` &dispatch, uint32\_t offset, uint32\_t max\_size=Maxu32)

## 5.243.1 Detailed Description

[SpatialTree](#) class

## 5.243.2 Member Function Documentation

## 5.243.2.1 create()

```
bool Tellusim::SpatialTree::create (
    const Device & device,
    Mode mode,
    RadixSort & sort,
    uint32_t size,
    uint32_t groups = 256,
    uint32_t regions = 1,
    Async * async = nullptr )
```

create spatial tree

## Parameters

<i>sort</i>	Radix sort.
<i>size</i>	<a href="#">Spatial</a> tree data size.
<i>groups</i>	<a href="#">Spatial</a> tree group size.
<i>regions</i>	Maximum number of multiple regions.

## 5.243.2.2 dispatch() [1/2]

```
bool Tellusim::SpatialTree::dispatch (
    Compute & compute,
    Hash hash,
    Buffer & nodes,
    uint32_t offset,
    uint32_t size,
    Flags flags = FlagNone )
```

dispatch inplace single spatial tree generation

## Parameters

<i>hash</i>	<a href="#">Spatial</a> tree hash mode.
<i>nodes</i>	<a href="#">Buffer</a> of spatial tree nodes.
<i>offset</i>	<a href="#">Spatial</a> tree nodes offset index.
<i>size</i>	Number of spatial elements.

5.243.2.3 `dispatch()` [2/2]

```
bool Tellusim::SpatialTree::dispatch (
    Compute & compute,
    Hash hash,
    Buffer & nodes,
    uint32_t count,
    const uint32_t * offsets,
    const uint32_t * sizes,
    Flags flags = FlagNone )
```

dispatch inplace multiple spatial tree generation

## Parameters

<i>hash</i>	<a href="#">Spatial</a> tree hash mode.
<i>nodes</i>	<a href="#">Buffer</a> of spatial tree nodes.
<i>count</i>	Number of regions to create.
<i>offsets</i>	<a href="#">Spatial</a> tree nodes offset index.
<i>sizes</i>	Number of spatial elements.

5.243.2.4 `dispatchIndirect()` [1/2]

```
bool Tellusim::SpatialTree::dispatchIndirect (
    Compute & compute,
    Hash hash,
    Buffer & nodes,
    Buffer & dispatch,
    uint32_t offset,
    uint32_t max_size = Maxu32 )
```

dispatch inplace single spatial tree generation

## Parameters

<i>hash</i>	<a href="#">Spatial</a> tree hash mode.
<i>nodes</i>	<a href="#">Buffer</a> of spatial tree nodes.
<i>dispatch</i>	Dispatch indirect buffer.
<i>offset</i>	Dispatch indirect buffer offset.
<i>max_size</i>	Maximum number of spatial elements.

5.243.2.5 `dispatchIndirect()` [2/2]

```
bool Tellusim::SpatialTree::dispatchIndirect (
    Compute & compute,
    Hash hash,
    Buffer & nodes,
    uint32_t count,
```



```

    Buffer & dispatch,
    uint32_t offset,
    uint32_t max_size = Maxu32 )

```

dispatch inplace multiple spatial tree generation

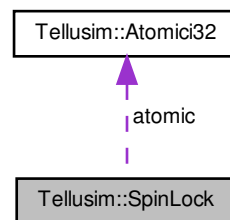
#### Parameters

<i>hash</i>	<a href="#">Spatial</a> tree hash mode.
<i>nodes</i>	<a href="#">Buffer</a> of spatial tree nodes.
<i>count</i>	Number of regions to create.
<i>dispatch</i>	Dispatch indirect buffer.
<i>offset</i>	Dispatch indirect buffer offset.
<i>max_size</i>	Maximum number of spatial elements.

## 5.244 Tellusim::SpinLock Struct Reference

```
#include <core/TellusimAtomic.h>
```

Collaboration diagram for Tellusim::SpinLock:



#### Public Member Functions

- **operator bool** ()
- void **clear** ()
- void **signal** ()
- void **lock** ()
- void **unlock** ()
- bool **check** ()
- int32\_t **get** ()

#### Public Attributes

- [Atomic32](#) **atomic**

### 5.244.1 Detailed Description

[SpinLock](#) class

## 5.245 Tellusim::Query::Statistics Struct Reference

statistics query

```
#include <platform/TellusimQuery.h>
```

### Public Attributes

- uint64\_t **num\_vertices**
- uint64\_t **num\_primitives**
- uint64\_t **vertex\_invocations**
- uint64\_t **control\_invocations**
- uint64\_t **evaluate\_invocations**
- uint64\_t **geometry\_invocations**
- uint64\_t **geometry\_primitives**
- uint64\_t **fragment\_invocations**
- uint64\_t **compute\_invocations**
- uint64\_t **clipping\_invocations**
- uint64\_t **clipping\_primitives**

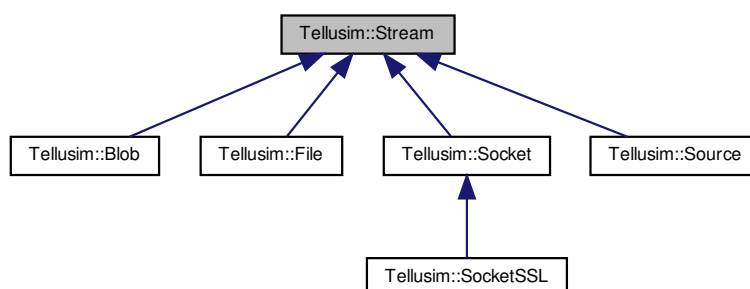
### 5.245.1 Detailed Description

statistics query

## 5.246 Tellusim::Stream Class Reference

```
#include <core/TellusimStream.h>
```

Inheritance diagram for Tellusim::Stream:



## Public Member Functions

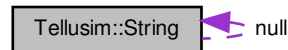
- [Stream](#) [move](#) ()  
*move stream*
- bool [isOpen](#) () const  
*stream status*
- bool [isMapped](#) () const
- bool [isAvailable](#) () const
- size\_t [getSize](#) () const
- [String](#) [getName](#) () const
- size\_t [tell](#) ()  
*stream position*
- bool [seek](#) (size\_t offset)
- bool [seekBack](#) (size\_t offset)
- bool [seekCur](#) (int64\_t offset)
- const uint8\_t \* [getData](#) () const  
*stream data*
- size\_t [read](#) (void \*dest, size\_t size)  
*read/write stream*
- size\_t [write](#) (const void \*src, size\_t size)
- bool [flush](#) ()
- bool [puts](#) (const char \*str)  
*unterminated strings*
- bool [puts](#) (const [String](#) &str)
- bool [vprintf](#) (const char \*format, va\_list args)
- bool [printf](#) (const char \*format,...) 1(2)
- template<class... List>  
bool [bool](#) [tprintf](#) (const char \*format, List... Args)
- [String](#) [gets](#) (bool \*status=nullptr)
- int8\_t [readi8](#) (bool \*status=nullptr)  
*8-bit integer numbers*
- bool [writei8](#) (int8\_t value)
- uint8\_t [readu8](#) (bool \*status=nullptr)
- bool [writeu8](#) (uint8\_t value)
- int16\_t [readi16](#) (bool \*status=nullptr)  
*16-bit integer numbers*
- bool [writei16](#) (int16\_t value)
- uint16\_t [readu16](#) (bool \*status=nullptr)
- bool [writeu16](#) (uint16\_t value)
- int32\_t [readi32](#) (bool \*status=nullptr)  
*32-bit integer numbers*
- bool [writei32](#) (int32\_t value)
- uint32\_t [readu32](#) (bool \*status=nullptr)
- bool [writeu32](#) (uint32\_t value)
- int64\_t [readi64](#) (bool \*status=nullptr)  
*64-bit integer numbers*
- bool [writei64](#) (int64\_t value)
- uint64\_t [readu64](#) (bool \*status=nullptr)
- bool [writeu64](#) (uint64\_t value)
- float32\_t [readf32](#) (bool \*status=nullptr)  
*32-bit floating-point numbers*
- bool [writef32](#) (float32\_t value)
- float64\_t [readf64](#) (bool \*status=nullptr)



## 5.247 Tellusim::String Class Reference

```
#include <core/TellusimString.h>
```

Collaboration diagram for Tellusim::String:



## Public Member Functions

- **String** (uint32\_t [size](#), char c=0)
- **String** (const char \*str, uint32\_t length=Maxu32)
- **String** (const wchar\_t \*str, uint32\_t length=Maxu32)
- **String** (const uint32\_t \*str, uint32\_t length=Maxu32)
- [String](#) & [reserve](#) (uint32\_t [size](#), bool discard=false)
- resize string*
- [String](#) & **resize** (uint32\_t [size](#), char c=0, bool [reserve](#)=false)
- void [release](#) ()
- clear string*
- void **clear** ()
- void [copy](#) (const char \*str, uint32\_t length=Maxu32)
- copy string*
- void **copy** (const wchar\_t \*str, uint32\_t length=Maxu32)
- void **copy** (const uint32\_t \*str, uint32\_t length=Maxu32)
- void **copy** (const [String](#) &string, uint32\_t length=Maxu32)
- [String](#) & **operator=** (const char \*str)
- [String](#) & [append](#) (char c)
- append string*
- [String](#) & **append** (const char \*str, uint32\_t length=Maxu32)
- [String](#) & **append** (const [String](#) &string, uint32\_t length=Maxu32)
- [String](#) & **operator+=** (char c)
- [String](#) & **operator+=** (const char \*str)
- [String](#) & **operator+=** (const [String](#) &string)
- [String](#) & **insert** (uint32\_t pos, const char \*str, uint32\_t length=Maxu32)
- insert string*
- [String](#) & **insert** (uint32\_t pos, const [String](#) &string, uint32\_t length=Maxu32)
- [String](#) & **removeBack** (uint32\_t length=1)
- remove string*
- [String](#) & **remove** (uint32\_t pos, uint32\_t length=1)
- [String](#) & **reverse** (uint32\_t pos=0, uint32\_t length=Maxu32)
- reverse string*
- uint32\_t [size](#) () const
- string info*
- bool **empty** () const
- **operator bool** () const

- char \* **get** ()
  - string data*
- const char \* **get** () const
- char & **get** (uint32\_t index)
- char **get** (uint32\_t index) const
- char & **operator[]** (uint32\_t index)
- char **operator[]** (uint32\_t index) const
- uint32\_t **find** (char c, uint32\_t pos=0) const
  - find ascii character*
- uint32\_t **rfind** (char c, uint32\_t pos=Maxu32) const
- uint32\_t **count** (char c, uint32\_t pos=0) const
- uint32\_t **find** (const char \*str, uint32\_t pos=0) const
  - find part of the string*
- uint32\_t **rfind** (const char \*str, uint32\_t pos=Maxu32) const
- uint32\_t **count** (const char \*str, uint32\_t pos=0) const
- bool **begins** (const char \*str, uint32\_t length=Maxu32, uint32\_t pos=0) const
  - compare begin of the string*
- bool **contains** (const char \*str, uint32\_t length=Maxu32, uint32\_t pos=0) const
  - contains the string*
- bool **match** (const char \*str, uint32\_t length=Maxu32, uint32\_t pos=0) const
  - matches the pattern*
- int32\_t **compare** (const char \*str, uint32\_t pos=0) const
  - compare strings*
- uint32\_t **distance** (const char \*str, bool scan=false, uint32\_t pos=0) const
  - distance between strings*
- const char \* **begin** () const
  - string iterators*
- const char \* **end** () const
- char **front** (uint32\_t index=0) const
  - string elements*
- char **back** (uint32\_t index=0) const
- char & **front** (uint32\_t index=0)
- char & **back** (uint32\_t index=0)
- **String substring** (uint32\_t pos, uint32\_t length=Maxu32) const
  - return part of the string*
- **String replace** (char before, char after, uint32\_t pos=0) const
  - replace part of the string*
- **String replace** (const char \*before, const char \*after, uint32\_t pos=0) const
- const Array< **String** > **split** (const char \*delimiters, uint32\_t length=Maxu32) const
  - split string*
- const Array< **String** > **split** (const **String** &delimiters, uint32\_t length=Maxu32) const
- **String extension** (const char \*extension) const
  - file name utils*
- **String extension** () const
- **String pathname** () const
- **String basename** () const
- **String dirname** () const
- **String capitalize** (const char \*delimiters=nullptr, const char \*spaces=nullptr) const
  - convert string case*
- **String lower** () const
- **String upper** () const
- uint32\_t **toUtf16** (wchar\_t \*d, uint32\_t length) const

- uint32\_t **toUtf32** (uint32\_t \*d, uint32\_t length) const
- uint32\_t **vscanf** (const char \*format, va\_list args) const  
*string scan function*
- uint32\_t **scanf** (const char \*format,...) const 1(2)
- uint32\_t **String** & **vprintf** (const char \*format, va\_list args)  
*string printf function*
- **String** & **printf** (const char \*format,...) 1(2)
- template<class... List>  
**String** **String** & **tprintf** (const char \*f, List... Args)
- int32\_t **toi32** (uint32\_t radix=10, uint32\_t pos=0) const
- int64\_t **toi64** (uint32\_t radix=10, uint32\_t pos=0) const
- uint32\_t **tou32** (uint32\_t radix=10, uint32\_t pos=0) const
- uint64\_t **tou64** (uint32\_t radix=10, uint32\_t pos=0) const
- float32\_t **tof32** (uint32\_t pos=0) const
- float64\_t **tof64** (uint32\_t pos=0) const
- uint32\_t **toHashu32** (uint32\_t pos=0) const  
*string to hash value*
- uint64\_t **toHashu64** (uint32\_t pos=0) const
- uint32\_t **toRGBAu8** (uint32\_t pos=0) const  
*string to color value*
- uint64\_t **toBytes** (uint32\_t pos=0, uint32\_t \*size=nullptr) const
- uint64\_t **toNumber** (uint32\_t pos=0, uint32\_t \*size=nullptr) const
- uint64\_t **toFrequency** (uint32\_t pos=0, uint32\_t \*size=nullptr) const
- float64\_t **toLength** (uint32\_t pos=0, uint32\_t \*size=nullptr) const

#### Static Public Member Functions

- static **String** **rename** (const char \*path, const char \*str)  
*relative file name*
- static uint32\_t **toUtf32** (const char \*str, uint32\_t &code)  
*string to unicode*
- static uint32\_t **fromUtf32** (**String** &d, uint32\_t code)  
*unicode to string*
- static **String** **fromUtf16** (const wchar\_t \*str, uint32\_t length=Maxu32)
- static **String** **fromUtf32** (const uint32\_t \*str, uint32\_t length=Maxu32)
- static **String** **fromUrl** (const char \*str, uint32\_t length=Maxu32)  
*url to string*
- static **String** **vformat** (const char \*format, va\_list args)  
*string format function*
- static **String** **format** (const char \*format,...) 1(1)
- template<class... List>  
static **String** static **String** **tformat** (const char \*format, List... Args)
- static **String** & **fromi32** (**String** &d, int32\_t value, uint32\_t radix=10)  
*value to string*
- static **String** & **fromi64** (**String** &d, int64\_t value, uint32\_t radix=10)
- static **String** & **fromu32** (**String** &d, uint32\_t value, uint32\_t radix=10)
- static **String** & **fromu64** (**String** &d, uint64\_t value, uint32\_t radix=10)
- static **String** & **fromf32** (**String** &d, float32\_t value, uint32\_t digits=6, bool compact=false, bool exponent=false)
- static **String** & **fromf64** (**String** &d, float64\_t value, uint32\_t digits=12, bool compact=false, bool exponent=false)
- static **String** **fromi32** (int32\_t value, uint32\_t radix=10)

- static [String](#) **fromi64** (int64\_t value, uint32\_t radix=10)
- static [String](#) **fromu32** (uint32\_t value, uint32\_t radix=10)
- static [String](#) **fromu64** (uint64\_t value, uint32\_t radix=10)
- static [String](#) **fromf32** (float32\_t value, uint32\_t digits=6, bool compact=false, bool exponent=false)
- static [String](#) **fromf64** (float64\_t value, uint32\_t digits=12, bool compact=false, bool exponent=false)
- static int32\_t **toi32** (const char \*str, uint32\_t radix=10, uint32\_t \*size=nullptr)

*string to value*

- static int64\_t **toi64** (const char \*str, uint32\_t radix=10, uint32\_t \*size=nullptr)
- static uint32\_t **tou32** (const char \*str, uint32\_t radix=10, uint32\_t \*size=nullptr)
- static uint64\_t **tou64** (const char \*str, uint32\_t radix=10, uint32\_t \*size=nullptr)
- static int32\_t **toi32** (const char \*str, uint32\_t \*size)
- static int64\_t **toi64** (const char \*str, uint32\_t \*size)
- static uint32\_t **tou32** (const char \*str, uint32\_t \*size)
- static uint64\_t **tou64** (const char \*str, uint32\_t \*size)
- static float32\_t **tof32** (const char \*str, uint32\_t \*size=nullptr)
- static float64\_t **tof64** (const char \*str, uint32\_t \*size=nullptr)
- static [String](#) **fromTime** (uint64\_t usec, uint32\_t digits=2)

*value convertors*

- static [String](#) **fromBytes** (uint64\_t bytes, uint32\_t digits=2)
- static [String](#) **fromNumber** (uint64\_t value, uint32\_t digits=2)
- static [String](#) **fromFrequency** (uint64\_t hz, uint32\_t digits=2)
- static [String](#) **fromLength** (float64\_t distance, uint32\_t digits=2)
- static [String](#) **fromAngle** (float64\_t angle, uint32\_t digits=2)
- static uint64\_t **toBytes** (const char \*str, uint32\_t \*size=nullptr)
- static uint64\_t **toNumber** (const char \*str, uint32\_t \*size=nullptr)
- static uint64\_t **toFrequency** (const char \*str, uint32\_t \*size=nullptr)
- static float64\_t **toLength** (const char \*str, uint32\_t \*size=nullptr)

#### Static Public Attributes

- static const [String](#) **null**

*empty string*

#### 5.247.1 Detailed Description

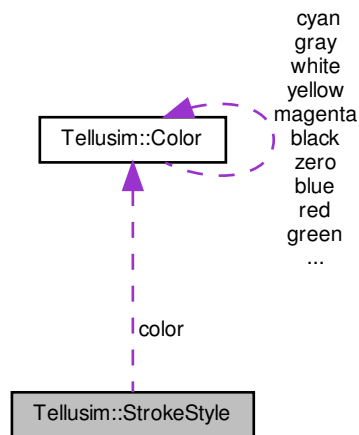
[String](#) class

#### 5.248 Tellusim::StrokeStyle Struct Reference

```
#include <interface/TellusimTypes.h>
```



Collaboration diagram for Tellusim::StrokeStyle:



#### Public Member Functions

- **StrokeStyle** (float32\_t width)
- **StrokeStyle** (const Color &color)
- **StrokeStyle** (float32\_t width, float32\_t offset)
- **StrokeStyle** (float32\_t width, const Color &color)
- **StrokeStyle** (float32\_t width, float32\_t offset, const Color &color)
- bool **isValid** () const  
*check style*
- **operator bool** () const

#### Public Attributes

- float32\_t **width** = 0.0f
- float32\_t **offset** = 0.0f
- Color **color** = Color::white

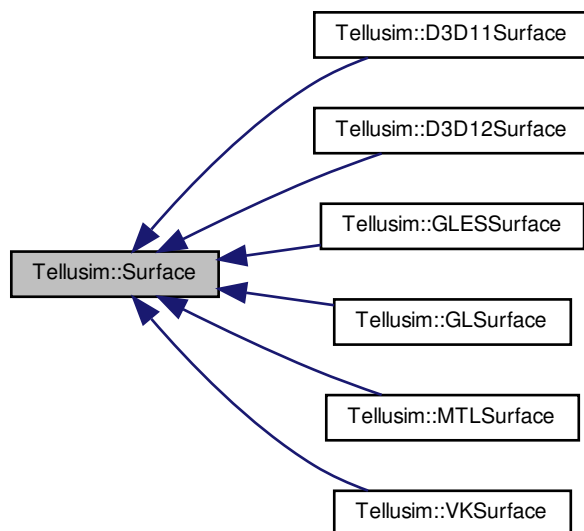
#### 5.248.1 Detailed Description

StrokeStyle

## 5.249 Tellusim::Surface Class Reference

```
#include <platform/TellusimSurface.h>
```

Inheritance diagram for Tellusim::Surface:



### Public Member Functions

- **Surface** ([Context](#) &context)
- **Surface** (Platform platform)
- Platform [getPlatform](#) () const  
*context platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*surface device index*
- void [setSize](#) (uint32\_t width, uint32\_t height)  
*surface size*
- uint32\_t **getWidth** () const
- uint32\_t **getHeight** () const
- void [setMultisample](#) (uint32\_t multisample)  
*surface multisample*
- uint32\_t **getMultisample** () const
- bool **hasMultisample** () const
- void [setColorLayer](#) (uint32\_t layer, uint32\_t layers)  
*surface layers*
- void **setDepthLayer** (uint32\_t layer, uint32\_t layers)
- uint32\_t **getColorLayer** () const
- uint32\_t **getDepthLayer** () const
- uint32\_t **getColorLayers** () const

- uint32\_t **getDepthLayers** () const
- bool **hasColorLayers** () const
- bool **hasDepthLayers** () const
- void **setColorFormat** (Format format)  
*surface formats*
- void **setDepthFormat** (Format format)
- Format **getColorFormat** () const
- Format **getDepthFormat** () const

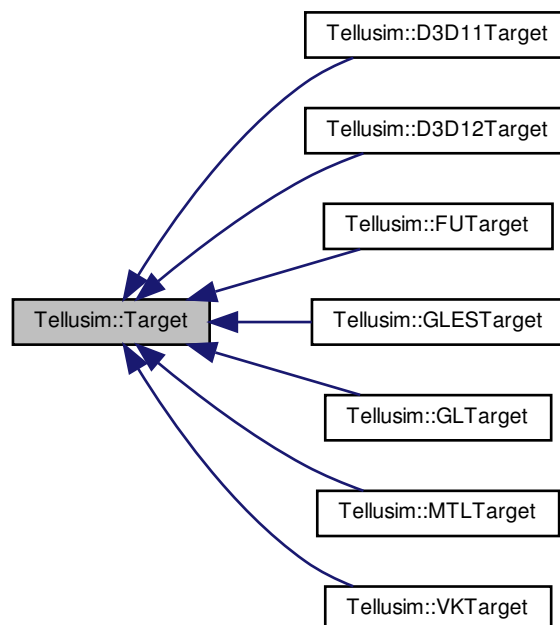
#### 5.249.1 Detailed Description

[Surface](#)

## 5.250 Tellusim::Target Class Reference

```
#include <platform/TellusimTarget.h>
```

Inheritance diagram for Tellusim::Target:



## Public Types

- enum [Operation](#) {  
**BeginLoad** = (1 << 0),  
**BeginClear** = (1 << 1),  
**BeginDiscard** = (1 << 2),  
**BeginMask** = (BeginLoad | BeginClear | BeginDiscard),  
**EndStore** = (1 << 3),  
**EndResolve** = (1 << 4),  
**EndDiscard** = (1 << 5),  
**EndMask** = (EndStore | EndResolve | EndDiscard),  
**OpNone** = 0,  
**OpLoad** = BeginLoad,  
**OpLoadStore** = (BeginLoad | EndStore),  
**OpClearStore** = (BeginClear | EndStore),  
**OpClearDiscard** = (BeginClear | EndDiscard),  
**OpDefault** = OpClearStore }

*[Target](#) operations.*

## Public Member Functions

- Platform [getPlatform](#) () const  
*target platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*target device index*
- bool [begin](#) ([Fence](#) &fence)  
*begin target*
- bool **begin** ()
- void [end](#) ([Fence](#) &fence)  
*end target*
- void **end** ()
- void [swap](#) ([Surface](#) &surface)  
*swap target*
- bool [isEnabled](#) () const  
*check target*
- Format [getColorFormat](#) (uint32\_t index=0) const  
*target format*
- Format **getDepthFormat** () const
- uint32\_t **getMultisample** () const
- bool **hasMultisample** () const
- bool **isFlipped** () const
- bool **isAtomic** () const
- uint32\_t [getWidth](#) () const  
*target dimension*
- uint32\_t **getHeight** () const
- uint32\_t **getDepth** () const
- uint32\_t **getFaces** () const
- uint32\_t **getLayers** () const
- uint32\_t **getMipmaps** () const
- void [setClearColor](#) (const [Color](#) &color)  
*color target*
- void **setClearColor** (uint32\_t index, const [Color](#) &color)

- void **setClearColor** (float32\_t r, float32\_t g, float32\_t b, float32\_t a)
- void **setClearColor** (uint32\_t index, float32\_t r, float32\_t g, float32\_t b, float32\_t a)
- void **setColorTexture** ([Texture](#) &texture, [Operation](#) op=OpDefault, const [Slice](#) &slice=[Slice](#)())
- void **setColorTexture** (uint32\_t index, [Texture](#) &texture, [Operation](#) op=OpDefault, const [Slice](#) &slice=[Slice](#)())
- void **setColorResolve** ([Texture](#) &texture, const [Slice](#) &slice=[Slice](#)())
- void **setColorResolve** (uint32\_t index, [Texture](#) &texture, const [Slice](#) &slice=[Slice](#)())
- uint32\_t **getNumTargets** () const
- const [Color](#) & **getClearColor** (uint32\_t index=0) const
- [Operation](#) **getColorOp** (uint32\_t index=0) const
- [Texture](#) **getColorTexture** (uint32\_t index) const
- [Texture](#) **getColorResolve** (uint32\_t index) const
- const [Slice](#) & **getColorTextureSlice** (uint32\_t index) const
- const [Slice](#) & **getColorResolveSlice** (uint32\_t index) const
- void **setClearDepth** (float32\_t depth, uint32\_t stencil=0)  
*depth target*
- void **setDepthTexture** ([Texture](#) &texture, [Operation](#) op=OpDefault, const [Slice](#) &slice=[Slice](#)())
- void **setDepthResolve** ([Texture](#) &texture, const [Slice](#) &slice=[Slice](#)())
- float32\_t **getClearDepth** () const
- uint32\_t **getClearStencil** () const
- [Operation](#) **getDepthOp** () const
- [Texture](#) **getDepthTexture** () const
- [Texture](#) **getDepthResolve** () const
- const [Slice](#) & **getDepthTextureSlice** () const
- const [Slice](#) & **getDepthResolveSlice** () const

### 5.250.1 Detailed Description

[Target](#) class

## 5.251 Tellusim::Async::Task Class Reference

[Task](#).

```
#include <core/TellusimAsync.h>
```

### Public Member Functions

- bool **operator==** (const [Task](#) &task) const  
*comparison operators*
- bool **operator!=** (const [Task](#) &task) const
- bool **empty** () const  
*task info*
- **operator bool** () const
- void **clear** ()  
*clear functions queue*
- void **cancel** ()  
*cancel functions queue*
- uint32\_t **index** ()  
*queue thread index*
- uint32\_t **size** () const

- number of queued functions*
- template<class Func >  
**Task & run** (const Func &func)  
  - run the function on the queue*
- template<class Func , class A0 >  
**Task & run** (const Func &func, A0 a0)
- template<class Func , class A0 , class A1 >  
**Task & run** (const Func &func, A0 a0, A1 a1)
- template<class Func , class A0 , class A1 , class A2 >  
**Task & run** (const Func &func, A0 a0, A1 a1, A2 a2)
- template<class Func , class A0 , class A1 , class A2 , class A3 >  
**Task & run** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3)
- template<class Func , class A0 , class A1 , class A2 , class A3 , class A4 >  
**Task & run** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3, A4 a4)
- template<class Func , class A0 , class A1 , class A2 , class A3 , class A4 , class A5 >  
**Task & run** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3, A4 a4, A5 a5)
- template<class Func , class A0 , class A1 , class A2 , class A3 , class A4 , class A5 , class A6 >  
**Task & run** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3, A4 a4, A5 a5, A6 a6)
- template<class Func , class A0 , class A1 , class A2 , class A3 , class A4 , class A5 , class A6 , class A7 >  
**Task & run** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3, A4 a4, A5 a5, A6 a6, A7 a7)
- template<class Func >  
**Task & run** (const Function< Func > &func)
- bool **check** (uint32\_t num=Maxu32) const  
  - check task completion status*
- bool **wait** (uint32\_t num=Maxu32) const  
  - waiting for the task completion*
- template<class Ret >  
const Ret & **get** (uint32\_t num=0) const  
  - function result*
- template<class Ret >  
Ret **getPtr** (uint32\_t num=0) const
- bool **getBool** (uint32\_t num=0) const

## Friends

- class **Async**

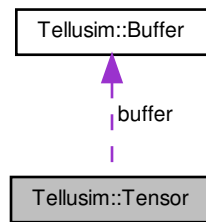
### 5.251.1 Detailed Description

**Task.**

### 5.252 Tellusim::Tensor Struct Reference

```
#include <parallel/TellusimTensorGraph.h>
```

Collaboration diagram for Tellusim::Tensor:



### Public Member Functions

- **Tensor** ([Buffer](#) \*buffer, size\_t offset=0)
- **Tensor** (Format format, uint32\_t width=0, uint32\_t height=1, uint32\_t depth=1, uint32\_t layers=1)
- **Tensor** ([Buffer](#) \*buffer, Format format, uint32\_t width=0, uint32\_t height=1, uint32\_t depth=1, uint32\_t layers=1)
- **Tensor** (const [Tensor](#) &t)
- **Tensor** (const [Tensor](#) &t, uint32\_t width, uint32\_t height=1, uint32\_t depth=1, uint32\_t layers=1)
- bool **isValid** () const  
*check tensor*
- **operator bool** () const
- uint32\_t **getSize** () const  
*tensor size*
- size\_t **getBytes** () const
- [Tensor](#) **setAxis** (uint32\_t axis) const  
*set operation parameters*
- [Tensor](#) **setKernel** (uint32\_t kernel) const
- [Tensor](#) **setStride** (uint32\_t stride) const
- [Tensor](#) **setPadding** (uint32\_t padding) const
- [Tensor](#) **setDilation** (uint32\_t dilation) const
- [Tensor](#) **setScaleBias** (float32\_t scale, float32\_t bias) const  
*set value parameters*
- [Tensor](#) **setScale** (float32\_t scale) const
- [Tensor](#) **setBias** (float32\_t bias) const

### Public Attributes

- [Buffer](#) \* **buffer** = nullptr
- Format **format** = FormatUnknown
- size\_t **offset** = 0
- uint32\_t **axis** = 0
- uint32\_t **kernel** = 2
- uint32\_t **stride** = 1
- uint32\_t **padding** = 0
- uint32\_t **dilation** = 1
- float32\_t **scale** = 1.0f

- float32\_t **bias** = 0.0f
- ```

union {
    struct {
        uint32_t width
        uint32_t height
        uint32_t depth
        uint32_t layers
    }
    uint32_t size [4]
};

```

#### 5.252.1 Detailed Description

#### Tensor

### 5.253 Tellusim::TensorGraph Class Reference

```
#include <parallel/TellusimTensorGraph.h>
```

#### Public Types

- enum [Operation](#) {  
**Clear** = 0,  
**Range**,  
**Copy**,  
**Cat**,  
**Transpose**,  
**MatMul**,  
**Mul**,  
**Mad**,  
**Div**,  
**Add**,  
**Conv**,  
**DeConv**,  
**BatchNorm**,  
**BatchMad**,  
**SoftMin**,  
**SoftMax**,  
**MaxPool**,  
**AvgPool**,  
**GELU**,  
**ReLU**,  
**SiLU**,  
**Sigm**,  
**Tanh**,  
**Sin**,  
**Cos**,  
**Exp**,  
**NumOperations** }  
*Graph operations.*



- enum **Flags** {
  - FlagNone** = 0,
  - FlagSizeQuery** = (1 << 0),
  - FlagFormatRf32** = (1 << 1),
  - FlagFormatRf16** = (1 << 2),
  - FlagTranspose** = (1 << 3),
  - FlagWrapClamp** = (1 << 4),
  - FlagWrapRepeat** = (1 << 5),
  - FlagReadScale** = (1 << 6),
  - FlagReadBias** = (1 << 7),
  - FlagConvert** = (1 << 8),
  - FlagKernel** = (1 << 9),
  - FlagGELU** = (1 << 10),
  - FlagReLU** = (1 << 11),
  - FlagSiLU** = (1 << 12),
  - FlagSigm** = (1 << 13),
  - FlagTanh** = (1 << 14),
  - FlagSin** = (1 << 15),
  - FlagCos** = (1 << 16),
  - FlagExp** = (1 << 17),
  - FlagFormat** = FlagFormatRf32 | FlagFormatRf16,
  - FlagWrap** = FlagWrapClamp | FlagWrapRepeat,
  - FlagRead** = FlagReadScale | FlagReadBias,
  - FlagUnit** = FlagGELU | FlagReLU | FlagSiLU,
  - FlagMath** = FlagSigm | FlagTanh | FlagSin | FlagCos | FlagExp,
  - FlagsAll** = FlagFormat | FlagTranspose | FlagWrap | FlagRead | FlagConvert | FlagKernel | FlagUnit | FlagMath }

*Graph flags.*

- enum **Masks** {
  - MaskNone** = 0,
  - MaskClear** = (1 << Clear),
  - MaskRange** = (1 << Range),
  - MaskCopy** = (1 << Copy),
  - MaskCat** = (1 << Cat),
  - MaskTranspose** = (1 << Transpose),
  - MaskMatMul** = (1 << MatMul),
  - MaskMul** = (1 << Mul),
  - MaskMad** = (1 << Mad),
  - MaskDiv** = (1 << Div),
  - MaskAdd** = (1 << Add),
  - MaskConv** = (1 << Conv),
  - MaskDeConv** = (1 << DeConv),
  - MaskBatchNorm** = (1 << BatchNorm),
  - MaskBatchMad** = (1 << BatchMad),
  - MaskSoftMin** = (1 << SoftMin),
  - MaskSoftMax** = (1 << SoftMax),
  - MaskMaxPool** = (1 << MaxPool),
  - MaskAvgPool** = (1 << AvgPool),
  - MaskGELU** = (1 << GELU),
  - MaskReLU** = (1 << ReLU),
  - MaskSiLU** = (1 << SiLU),
  - MaskSigm** = (1 << Sigm),
  - MaskTanh** = (1 << Tanh),
  - MaskSin** = (1 << Sin),
  - MaskCos** = (1 << Cos),
  - MaskExp** = (1 << Exp),
  - MasksAll** = (1 << NumOperations) - 1 }

*Graph masks.*

## Public Member Functions

- void **clear** ()  
*clear graph*
- bool **isCreated** () const  
*check graph*
- bool **create** (const **Device** &device, **Flags** flags=FlagsAll, **Masks** masks=MasksAll, **Async** \*async=nullptr)  
*create graph*
- bool **dispatch** (**Compute** &compute, **Operation** op, const **Tensor** &dest, **Flags** flags=FlagNone) const
- bool **dispatch** (**Compute** &compute, **Operation** op, **Tensor** &dest, const **Tensor** &src\_0, **Flags** flags=FlagNone) const
- bool **dispatch** (**Compute** &compute, **Operation** op, **Tensor** &dest, const **Tensor** &src\_0, const **Tensor** &src\_1, **Flags** flags=FlagNone) const
- bool **dispatch** (**Compute** &compute, **Operation** op, **Tensor** &dest, const **Tensor** &src\_0, const **Tensor** &src\_1, const **Tensor** &src\_2, **Flags** flags=FlagNone) const
- bool **dispatch** (**Compute** &compute, const **Tensor** &dest, **Texture** &src, const **Region** &region, const **Slice** &slice=**Slice**()) const  
*dispatch Texture to Tensor*
- bool **dispatch** (**Compute** &compute, const **Tensor** &dest, **Texture** &src, const **Slice** &slice=**Slice**()) const
- bool **dispatch** (**Compute** &compute, **Texture** &dest, const **Tensor** &src, const **Region** &region, const **Slice** &slice=**Slice**()) const  
*dispatch Tensor to Texture*
- bool **dispatch** (**Compute** &compute, **Texture** &dest, const **Tensor** &src, const **Slice** &slice=**Slice**()) const

## 5.253.1 Detailed Description

**TensorGraph** class

## 5.253.2 Member Function Documentation

5.253.2.1 **dispatch**()

```
bool Tellusim::TensorGraph::dispatch (
    Compute & compute,
    Operation op,
    const Tensor & dest,
    Flags flags = FlagNone ) const
```

dispatch **Tensor** operation

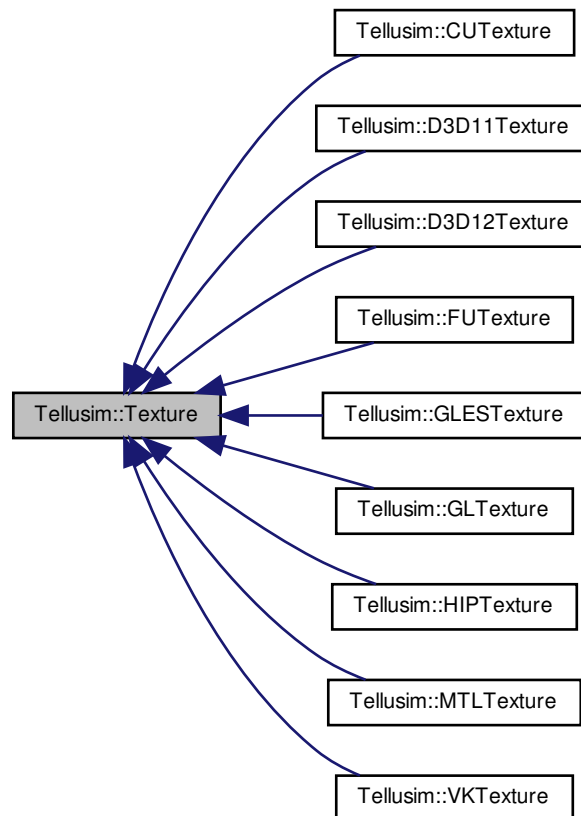
## Parameters

|              |                       |
|--------------|-----------------------|
| <i>op</i>    | Graph operation.      |
| <i>flags</i> | Operation flags.      |
| <i>dest</i>  | Destination tensor.   |
| <i>src</i>   | <b>Source</b> tensor. |

## 5.254 Tellusim::Texture Class Reference

```
#include <platform/TellusimTexture.h>
```

Inheritance diagram for Tellusim::Texture:



## Public Types

- enum `Flags` {
  - `FlagNone` = 0,
  - `FlagRead` = (1 << 0),
  - `FlagWrite` = (1 << 1),
  - `FlagTarget` = (1 << 2),
  - `FlagBuffer` = (1 << 3),
  - `FlagSource` = (1 << 4),
  - `FlagSparse` = (1 << 5),
  - `FlagShared` = (1 << 6),
  - `FlagExtern` = (1 << 7),
  - `FlagInterop` = (1 << 8),
  - `FlagSurface` = (1 << 9),
  - `FlagMutable` = (1 << 10),
  - `FlagMipmaps` = (1 << 11),

```

FlagGenerate = (1 << 12),
FlagFormatNorm = (1 << 13),
FlagFormatSRGB = (1 << 14),
FlagFormatSigned = (1 << 15),
FlagMultisample2 = (1 << 16),
FlagMultisample4 = (1 << 17),
FlagMultisample8 = (1 << 18),
FlagClearOne = (1 << 19),
FlagClearZero = (1 << 20),
FlagClearNormal = (1 << 21),
FlagMultisample = (FlagMultisample2 | FlagMultisample4 | FlagMultisample8),
DefaultFlags = FlagNone,
NumFlags = 22 }

```

*Texture flags.*

## Public Member Functions

- Platform [getPlatform](#) () const  
*texture platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*texture device index*
- void [clear](#) ()  
*clear texture*
- bool [isCreated](#) () const  
*check texture*
- void [setName](#) (const char \*name)  
*texture name*
- [String](#) [getName](#) () const
- bool [create](#) (Type type, Format format, const [Size](#) &size, uint32\_t layers, [Flags](#) flags=DefaultFlags)  
*create texture*
- bool [create2D](#) (Format format, uint32\_t size, [Flags](#) flags=DefaultFlags)
- bool [create3D](#) (Format format, uint32\_t size, [Flags](#) flags=DefaultFlags)
- bool [createCube](#) (Format format, uint32\_t size, [Flags](#) flags=DefaultFlags)
- bool [create2D](#) (Format format, uint32\_t width, uint32\_t height, [Flags](#) flags=DefaultFlags)
- bool [create3D](#) (Format format, uint32\_t width, uint32\_t height, uint32\_t depth, [Flags](#) flags=DefaultFlags)
- bool [create2D](#) (Format format, uint32\_t width, uint32\_t height, uint32\_t layers, [Flags](#) flags=DefaultFlags)
- bool [createCube](#) (Format format, uint32\_t size, uint32\_t layers, [Flags](#) flags=DefaultFlags)
- Type [getType](#) () const  
*texture type*
- const char \* **getTypeName** () const
- bool **is2DType** () const
- bool **is3DType** () const
- bool **isCubeType** () const
- Format [getFormat](#) () const  
*texture format*
- const char \* **getFormatName** () const
- bool **isColorFormat** () const
- bool **isDepthFormat** () const
- bool **isPixelFormat** () const
- bool **isPlainFormat** () const
- bool **isMixedFormat** () const
- bool **isBlockFormat** () const

- bool **isStencilFormat** () const
- bool **isNormFormat** () const
- bool **isSRGBFormat** () const
- bool **isFloatFormat** () const
- bool **isSignedFormat** () const
- bool **isUnsignedFormat** () const
- bool **isIntegerFormat** () const
- bool **isi8Format** () const
- bool **isu8Format** () const
- bool **is8BitFormat** () const
- bool **isi16Format** () const
- bool **isu16Format** () const
- bool **isf16Format** () const
- bool **is16BitFormat** () const
- bool **isi32Format** () const
- bool **isu32Format** () const
- bool **isf32Format** () const
- bool **is32BitFormat** () const
- bool **isi64Format** () const
- bool **isu64Format** () const
- bool **isf64Format** () const
- bool **is64BitFormat** () const
- bool **isBC15Format** () const
- bool **isBC67Format** () const
- bool **isETC2Format** () const
- bool **isASTCFormat** () const
- uint32\_t **getComponents** () const
- uint32\_t **getPixelSize** () const
- uint32\_t **getBlockSize** () const
- uint32\_t **getBlockWidth** () const
- uint32\_t **getBlockHeight** () const
- [Flags](#) **getFlags** () const
- texture flags*
- bool **hasFlag** ([Flags](#) flags) const
- bool **hasFlags** ([Flags](#) flags) const
- [String](#) **getFlagsName** () const
- uint32\_t **getMultisample** () const
- texture multisample*
- bool **hasMultisample** () const
- uint32\_t **getWidth** () const
- texture dimension*
- uint32\_t **getHeight** () const
- uint32\_t **getDepth** () const
- uint32\_t **getFaces** () const
- uint32\_t **getLayers** () const
- uint32\_t **getMipmaps** () const
- uint32\_t **findMipmap** (const [Size](#) &size) const
- uint32\_t **getWidth** (uint32\_t mipmap) const
- uint32\_t **getHeight** (uint32\_t mipmap) const
- uint32\_t **getDepth** (uint32\_t mipmap) const
- bool **hasFaces** () const
- bool **hasLayers** () const
- bool **hasMipmaps** () const
- [Size](#) **getSize** () const

- [Region](#) **getRegion** () const
- [Slice](#) **getSlice** () const
- [Size](#) **getSize** (uint32\_t mipmap) const
- [Region](#) **getRegion** (uint32\_t mipmap) const
- [Slice](#) **getSlice** (uint32\_t mipmap) const
- uint32\_t [getTileWidth](#) () const  
*sparse texture dimension*
- uint32\_t **getTileHeight** () const
- uint32\_t **getTileDepth** () const
- uint32\_t **getTileMipmaps** () const
- [Size](#) **getTileSize** () const
- [String](#) **getDescription** () const  
*texture description*
- size\_t [getMemory](#) () const  
*memory usage*

#### Static Public Member Functions

- static const char \* **getTypeName** (Type type)

#### 5.254.1 Detailed Description

[Texture](#) class

#### 5.255 Tellusim::TextureTable Class Reference

```
#include <platform/TellusimTexture.h>
```

#### Public Member Functions

- Platform [getPlatform](#) () const  
*table platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*table device index*
- void [clear](#) ()  
*clear table*
- bool [isCreated](#) () const  
*check table*
- void [setName](#) (const char \*name)  
*table name*
- [String](#) **getName** () const
- bool [create](#) (Texture::Type type, uint32\_t size)  
*create table*
- Texture::Type [getType](#) () const  
*table type*
- const char \* **getTypeName** () const
- uint32\_t [getSize](#) () const  
*table textures*
- [Texture](#) **get** (uint32\_t index) const
- bool **isOwner** (uint32\_t index) const
- size\_t [getMemory](#) () const  
*memory usage*

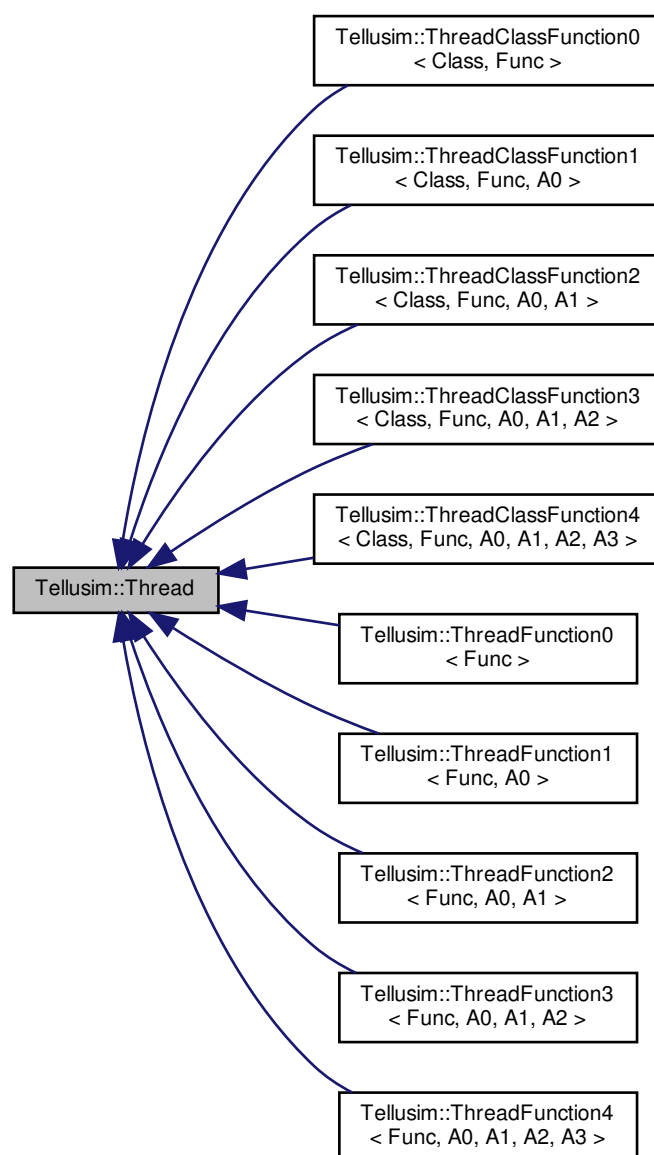
## 5.255.1 Detailed Description

[TextureTable](#) class

## 5.256 Tellusim::Thread Class Reference

```
#include <core/TellusimThread.h>
```

Inheritance diagram for Tellusim::Thread:



### Public Member Functions

- bool **run** (uint32\_t stack=1024 \*1024)  
*run the thread*
- bool **stop** (bool **wait**=false)  
*stop the thread*
- bool **wait** ()  
*wait for the signal*
- bool **signal** ()  
*wake up the thread*
- bool **terminate** ()  
*terminate the thread*
- bool **isRunning** () const  
*thread status*
- bool **isStopped** () const
- bool **isWaiting** () const

### Protected Member Functions

- virtual void **process** ()  
*thread process*

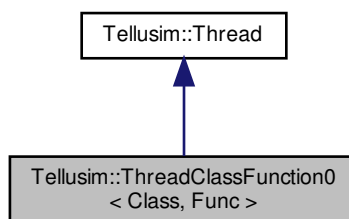
#### 5.256.1 Detailed Description

Hardware thread

### 5.257 Tellusim::ThreadClassFunction0< Class, Func > Class Template Reference

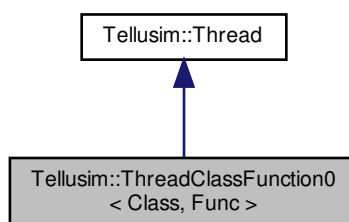
```
#include <core/TellusimThread.h>
```

Inheritance diagram for Tellusim::ThreadClassFunction0< Class, Func >:





Collaboration diagram for Tellusim::ThreadClassFunction0< Class, Func >:



#### Public Member Functions

- **ThreadClassFunction0** (Class \*c, const Func &func)

#### Protected Member Functions

- virtual void `process` ()  
*thread process*

#### Protected Attributes

- Class \* **c**
- Func **func**

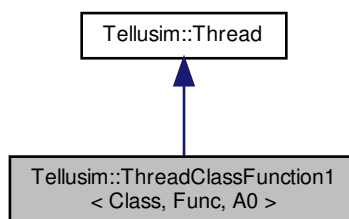
#### 5.257.1 Detailed Description

```
template<class Class, class Func>
class Tellusim::ThreadClassFunction0< Class, Func >
```

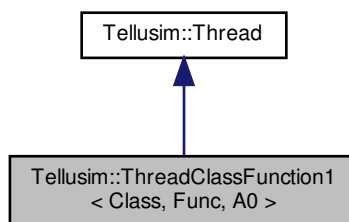
`Thread` class member function

#### 5.258 Tellusim::ThreadClassFunction1< Class, Func, A0 > Class Template Reference

Inheritance diagram for Tellusim::ThreadClassFunction1< Class, Func, A0 >:



Collaboration diagram for Tellusim::ThreadClassFunction1 < Class, Func, A0 >:



#### Public Member Functions

- **ThreadClassFunction1** (Class \*c, const Func &func, A0 a0)

#### Protected Member Functions

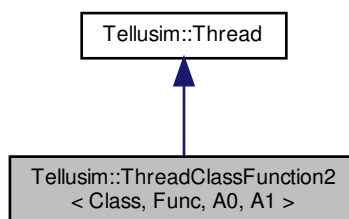
- virtual void [process](#) ()  
*thread process*

#### Protected Attributes

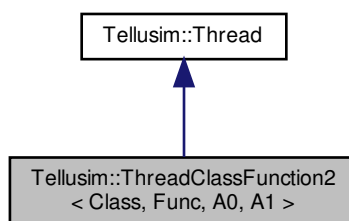
- Class \* **c**
- Func **func**
- A0 **a0**

### 5.259 Tellusim::ThreadClassFunction2< Class, Func, A0, A1 > Class Template Reference

Inheritance diagram for Tellusim::ThreadClassFunction2< Class, Func, A0, A1 >:



Collaboration diagram for Tellusim::ThreadClassFunction2< Class, Func, A0, A1 >:



#### Public Member Functions

- **ThreadClassFunction2** (Class \*c, const Func &func, A0 a0, A1 a1)

#### Protected Member Functions

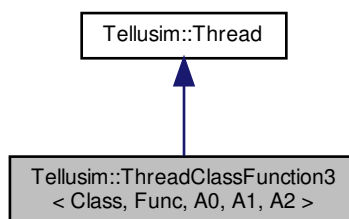
- virtual void `process` ()  
*thread process*

#### Protected Attributes

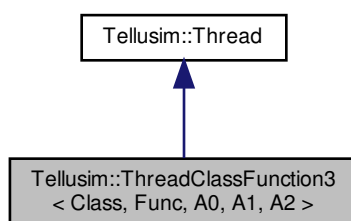
- Class \* **c**
- Func **func**
- A0 **a0**
- A1 **a1**

## 5.260 Tellusim::ThreadClassFunction3< Class, Func, A0, A1, A2 > Class Template Reference

Inheritance diagram for Tellusim::ThreadClassFunction3< Class, Func, A0, A1, A2 >:



Collaboration diagram for `Tellusim::ThreadClassFunction3< Class, Func, A0, A1, A2 >`:



#### Public Member Functions

- **ThreadClassFunction3** (`Class *c`, `const Func &func`, `A0 a0`, `A1 a1`, `A2 a2`)

#### Protected Member Functions

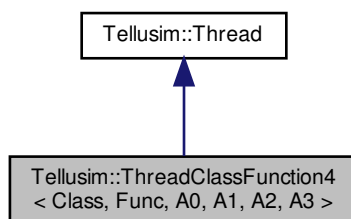
- virtual void `process` ()  
*thread process*

#### Protected Attributes

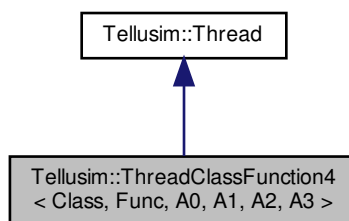
- `Class * c`
- `Func func`
- `A0 a0`
- `A1 a1`
- `A2 a2`

### 5.261 `Tellusim::ThreadClassFunction4< Class, Func, A0, A1, A2, A3 >` Class Template Reference

Inheritance diagram for `Tellusim::ThreadClassFunction4< Class, Func, A0, A1, A2, A3 >`:



Collaboration diagram for Tellusim::ThreadClassFunction4< Class, Func, A0, A1, A2, A3 >:



#### Public Member Functions

- **ThreadClassFunction4** (Class \*c, const Func &func, A0 a0, A1 a1, A2 a2, A3 a3)

#### Protected Member Functions

- virtual void `process` ()  
*thread process*

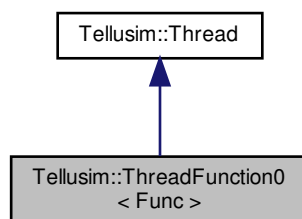
#### Protected Attributes

- Class \* **c**
- Func **func**
- A0 **a0**
- A1 **a1**
- A2 **a2**
- A3 **a3**

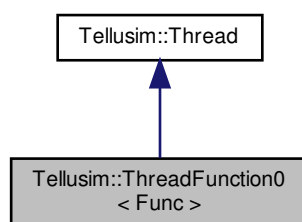
## 5.262 Tellusim::ThreadFunction0< Func > Class Template Reference

```
#include <core/TellusimThread.h>
```

Inheritance diagram for Tellusim::ThreadFunction0< Func >:



Collaboration diagram for Tellusim::ThreadFunction0< Func >:



#### Public Member Functions

- **ThreadFunction0** (const Func &func)

#### Protected Member Functions

- virtual void `process` ()  
*thread process*

#### Protected Attributes

- Func **func**

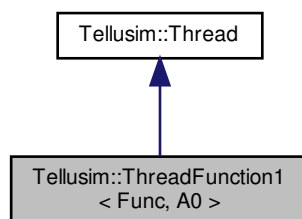
#### 5.262.1 Detailed Description

```
template<class Func>
class Tellusim::ThreadFunction0< Func >
```

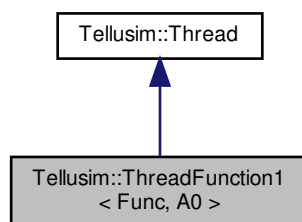
`Thread` function

#### 5.263 Tellusim::ThreadFunction1< Func, A0 > Class Template Reference

Inheritance diagram for Tellusim::ThreadFunction1< Func, A0 >:



Collaboration diagram for Tellusim::ThreadFunction1< Func, A0 >:



#### Public Member Functions

- **ThreadFunction1** (const Func &func, A0 a0)

#### Protected Member Functions

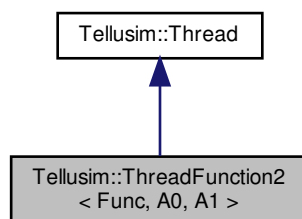
- virtual void `process` ()  
*thread process*

#### Protected Attributes

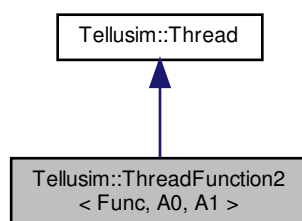
- Func **func**
- A0 **a0**

## 5.264 Tellusim::ThreadFunction2< Func, A0, A1 > Class Template Reference

Inheritance diagram for Tellusim::ThreadFunction2< Func, A0, A1 >:



Collaboration diagram for Tellusim::ThreadFunction2< Func, A0, A1 >:



#### Public Member Functions

- **ThreadFunction2** (const Func &func, A0 a0, A1 a1)

#### Protected Member Functions

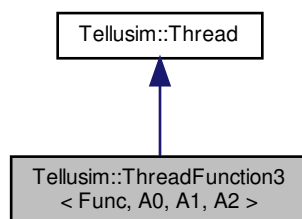
- virtual void [process](#) ()  
*thread process*

#### Protected Attributes

- Func **func**
- A0 **a0**
- A1 **a1**

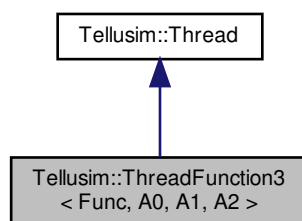
### 5.265 Tellusim::ThreadFunction3< Func, A0, A1, A2 > Class Template Reference

Inheritance diagram for Tellusim::ThreadFunction3< Func, A0, A1, A2 >:





Collaboration diagram for Tellusim::ThreadFunction3< Func, A0, A1, A2 >:



#### Public Member Functions

- **ThreadFunction3** (const Func &func, A0 a0, A1 a1, A2 a2)

#### Protected Member Functions

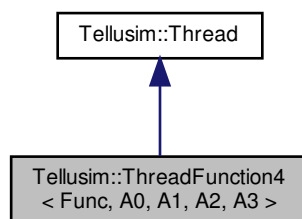
- virtual void `process` ()  
*thread process*

#### Protected Attributes

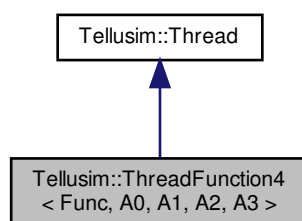
- Func **func**
- A0 **a0**
- A1 **a1**
- A2 **a2**

## 5.266 Tellusim::ThreadFunction4< Func, A0, A1, A2, A3 > Class Template Reference

Inheritance diagram for Tellusim::ThreadFunction4< Func, A0, A1, A2, A3 >:



Collaboration diagram for Tellusim::ThreadFunction4< Func, A0, A1, A2, A3 >:



#### Public Member Functions

- **ThreadFunction4** (const Func &func, A0 a0, A1 a1, A2 a2, A3 a3)

#### Protected Member Functions

- virtual void `process` ()  
*thread process*

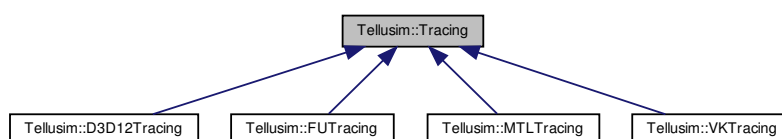
#### Protected Attributes

- Func **func**
- A0 **a0**
- A1 **a1**
- A2 **a2**
- A3 **a3**

## 5.267 Tellusim::Tracing Class Reference

```
#include <platform/TellusimTracing.h>
```

Inheritance diagram for Tellusim::Tracing:



## Classes

- struct [BuildIndirect](#)  
*build indirect parameters*
- struct [Instance](#)  
*tracing instance*

## Public Types

- enum [Flags](#) {  
    **FlagNone** = 0,  
    **FlagInfo** = (1 << 0),  
    **FlagUpdate** = (1 << 1),  
    **FlagCompact** = (1 << 2),  
    **FlagTransparent** = (1 << 3),  
    **FlagFastBuild** = (1 << 4),  
    **FlagFastTrace** = (1 << 5),  
    **DefaultFlags** = FlagNone,  
    **NumFlags** = 6 }  
    [Tracing](#) flags.
- enum { **InstanceSize** = 64 }  
    *instance size*

## Public Member Functions

- Platform [getPlatform](#) () const  
    *tracing platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
    *tracing device index*
- void [clear](#) ()  
    *clear tracing*
- bool [isCreated](#) () const  
    *check tracing*
- bool **isBuilt** () const
- void [setName](#) (const char \*name)  
    *tracing name*
- [String](#) **getName** () const
- bool [create](#) (Type type, [Flags](#) flags=DefaultFlags)  
    *create tracing*
- Type [getType](#) () const  
    *tracing type*
- const char \* **getTypeName** () const
- bool **isInstanceType** () const
- bool **isTriangleType** () const
- bool **isBoundType** () const
- bool **isGeometryType** () const
- void [setParameters](#) (const [Tracing](#) &tracing)  
    *tracing parameters*
- uint32\_t [getNumGeometries](#) () const  
    *tracing geometries*
- void [setInstanceBuffer](#) (uint32\_t num\_instances, [Buffer](#) &buffer, size\_t offset=0)

*instance buffer*

- void **setInstanceBuffer** ([Buffer](#) &buffer, size\_t offset=0)
- void **setNumInstances** (uint32\_t num\_instances)
- uint32\_t **getNumInstances** () const
- [Buffer](#) **getInstanceBuffer** () const
- size\_t **getInstanceOffset** () const
- void **setIndirectBuffer** ([Buffer](#) &buffer, size\_t offset=0)

*indirect buffer*

- [Buffer](#) **getIndirectBuffer** () const
- size\_t **getIndirectOffset** () const
- uint32\_t **addVertexBuffer** (uint32\_t num\_vertices, Format format, size\_t stride, [Buffer](#) buffer=[Buffer::null](#), size\_t offset=0)

*vertex buffers*

- void **setVertexBuffer** (uint32\_t index, uint32\_t num\_vertices, [Buffer](#) &buffer, size\_t offset=0)
- void **setVertexBuffer** (uint32\_t index, [Buffer](#) &buffer, size\_t offset=0)
- void **setNumVertices** (uint32\_t index, uint32\_t num\_vertices)
- uint32\_t **getNumVertices** (uint32\_t index) const
- Format **getVertexFormat** (uint32\_t index) const
- uint32\_t **getVertexStride** (uint32\_t index) const
- [Buffer](#) **getVertexBuffer** (uint32\_t index) const
- size\_t **getVertexOffset** (uint32\_t index) const
- uint32\_t **addIndexBuffer** (uint32\_t num\_indices, Format format, [Buffer](#) buffer=[Buffer::null](#), size\_t offset=0)

*index buffers*

- void **setIndexBuffer** (uint32\_t index, uint32\_t num\_indices, [Buffer](#) &buffer, size\_t offset=0)
- void **setIndexBuffer** (uint32\_t index, [Buffer](#) &buffer, size\_t offset=0)
- void **setNumIndices** (uint32\_t index, uint32\_t num\_indices)
- uint32\_t **getNumIndices** (uint32\_t index) const
- Format **getIndexFormat** (uint32\_t index) const
- [Buffer](#) **getIndexBuffer** (uint32\_t index) const
- size\_t **getIndexOffset** (uint32\_t index) const
- uint32\_t **addBoundBuffer** (uint32\_t num\_bounds, size\_t stride, [Buffer](#) buffer=[Buffer::null](#), size\_t offset=0)

*bound buffers*

- void **setBoundBuffer** (uint32\_t index, uint32\_t num\_bounds, [Buffer](#) &buffer, size\_t offset=0)
- void **setBoundBuffer** (uint32\_t index, [Buffer](#) &buffer, size\_t offset=0)
- void **setNumBounds** (uint32\_t index, uint32\_t num\_bounds)
- uint32\_t **getNumBounds** (uint32\_t index) const
- uint32\_t **getBoundStride** (uint32\_t index) const
- [Buffer](#) **getBoundBuffer** (uint32\_t index) const
- size\_t **getBoundOffset** (uint32\_t index) const
- [String](#) **getDescription** () const

*tracing description*

- uint64\_t **getTracingAddress** () const

*tracing address*

- size\_t **getBuildSize** () const

*scratch buffer size*

- size\_t **getUpdateSize** () const
- size\_t **getMemory** () const

*memory usage***Static Public Member Functions**

- static const char \* **getTypeName** (Type type)

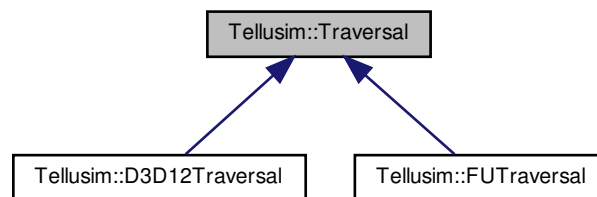
## 5.267.1 Detailed Description

[Tracing](#) class

## 5.268 Tellusim::Traversal Class Reference

```
#include <platform/TellusimTraversal.h>
```

Inheritance diagram for Tellusim::Traversal:



## Public Member Functions

- Platform [getPlatform](#) () const  
*traversal platform*
- const char \* [getPlatformName](#) () const
- uint32\_t [getIndex](#) () const  
*traversal device index*
- void [clear](#) ()  
*clear traversal*
- bool [isCreated](#) () const  
*check traversal*
- void [setName](#) (const char \*name)  
*traversal name*
- [String](#) [getName](#) () const
- bool [create](#) ()  
*create traversal*
- void [setParameters](#) (const [Traversal](#) &traversal)  
*traversal parameters*
- bool [saveState](#) ([Stream](#) &stream) const
- void [addShader](#) ([Shader](#) &shader, bool owner=false)  
*shader pointers*
- [Shader](#) [getRayGenShader](#) () const
- bool [loadShader](#) (Shader::Type type, const char \*name, const char \*format,...) 1(4)  
*load shaders*
- bool [loadShaderGLSL](#) (Shader::Type type, const char \*name, const char \*format,...) 1(4)
- bool [loadShader](#) (Shader::Type type, const char \*name, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)

- bool **loadShaderGLSL** (Shader::Type type, const char \*name, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
- bool **loadShaderSPIRV** (Shader::Type type, const char \*name)
- bool **createShader** (Shader::Type type, const char \*src, const char \*format,...) 1(4)
  - create shaders*
  - bool **createShaderGLSL** (Shader::Type type, const char \*src, const char \*format,...) 1(4)
  - bool **createShader** (Shader::Type type, const char \*src, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
  - bool **createShaderGLSL** (Shader::Type type, const char \*src, const [String](#) &macros=[String::null](#), const char \*\*includes=nullptr, uint32\_t size=0)
  - bool **createShaderSPIRV** (Shader::Type type, const Array< uint32\_t > &data)
  - uint32\_t **addSampler** ([Shader::Mask](#) mask)
    - sampler parameters*
    - uint32\_t **getNumSamplers** () const
    - [Traversal](#) & **setSamplerOffset** (uint32\_t offset)
    - uint32\_t **getSamplerOffset** () const
    - [Traversal](#) & **setSamplerMask** (uint32\_t index, [Shader::Mask](#) mask)
    - [Shader::Mask](#) **getSamplerMask** (uint32\_t index) const
    - [Traversal](#) & **setSamplerMasks** (uint32\_t index, uint32\_t num, [Shader::Mask](#) mask, bool array=false)
    - [Shader::Mask](#) **getSamplerMasks** (uint32\_t index, uint32\_t num) const
    - [Traversal](#) & **setSamplerArray** (uint32\_t index, uint32\_t num, bool array)
    - uint32\_t **getSamplerArray** (uint32\_t index) const
    - uint32\_t **addTexture** ([Shader::Mask](#) mask)
      - texture parameters*
      - uint32\_t **getNumTextures** () const
      - [Traversal](#) & **setTextureOffset** (uint32\_t offset)
      - uint32\_t **getTextureOffset** () const
      - [Traversal](#) & **setTextureMask** (uint32\_t index, [Shader::Mask](#) mask)
      - [Shader::Mask](#) **getTextureMask** (uint32\_t index) const
      - [Traversal](#) & **setTextureMasks** (uint32\_t index, uint32\_t num, [Shader::Mask](#) mask, bool array=false)
      - [Shader::Mask](#) **getTextureMasks** (uint32\_t index, uint32\_t num) const
      - [Traversal](#) & **setTextureArray** (uint32\_t index, uint32\_t num, bool array)
      - uint32\_t **getTextureArray** (uint32\_t index) const
      - uint32\_t **addSurface** ([Shader::Mask](#) mask)
        - surface parameters*
        - uint32\_t **getNumSurfaces** () const
        - [Traversal](#) & **setSurfaceOffset** (uint32\_t offset)
        - uint32\_t **getSurfaceOffset** () const
        - [Traversal](#) & **setSurfaceMask** (uint32\_t index, [Shader::Mask](#) mask)
        - [Shader::Mask](#) **getSurfaceMask** (uint32\_t index) const
        - [Traversal](#) & **setSurfaceMasks** (uint32\_t index, uint32\_t num, [Shader::Mask](#) mask, bool array=false)
        - [Shader::Mask](#) **getSurfaceMasks** (uint32\_t index, uint32\_t num) const
        - [Traversal](#) & **setSurfaceArray** (uint32\_t index, uint32\_t num, bool array)
        - uint32\_t **getSurfaceArray** (uint32\_t index) const
        - uint32\_t **addUniform** ([Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
          - uniform parameters*
          - uint32\_t **getNumUniforms** () const
          - [Traversal](#) & **setUniformOffset** (uint32\_t offset)
          - uint32\_t **getUniformOffset** () const
          - [Traversal](#) & **setUniformMask** (uint32\_t index, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
          - [Shader::Mask](#) **getUniformMask** (uint32\_t index) const
          - [Traversal](#) & **setUniformMasks** (uint32\_t index, uint32\_t num, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
          - [Shader::Mask](#) **getUniformMasks** (uint32\_t index, uint32\_t num) const

- [Traversal](#) & **setUniformFlags** (uint32\_t index, BindFlags flags)
- BindFlags **getUniformFlags** (uint32\_t index) const
- uint32\_t **addStorage** ([Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
- storage parameters*
- uint32\_t **getNumStorages** () const
- [Traversal](#) & **setStorageOffset** (uint32\_t offset)
- uint32\_t **getStorageOffset** () const
- [Traversal](#) & **setStorageMask** (uint32\_t index, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
- [Shader::Mask](#) **getStorageMask** (uint32\_t index) const
- [Traversal](#) & **setStorageMasks** (uint32\_t index, uint32\_t num, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
- [Shader::Mask](#) **getStorageMasks** (uint32\_t index, uint32\_t num) const
- [Traversal](#) & **setStorageFlags** (uint32\_t index, BindFlags flags)
- BindFlags **getStorageFlags** (uint32\_t index) const
- uint32\_t **addTracing** ([Shader::Mask](#) mask)
- tracing parameters*
- uint32\_t **getNumTracings** () const
- [Traversal](#) & **setTracingOffset** (uint32\_t offset)
- uint32\_t **getTracingOffset** () const
- [Traversal](#) & **setTracingMask** (uint32\_t index, [Shader::Mask](#) mask)
- [Shader::Mask](#) **getTracingMask** (uint32\_t index) const
- [Traversal](#) & **setTracingMasks** (uint32\_t index, uint32\_t num, [Shader::Mask](#) mask)
- [Shader::Mask](#) **getTracingMasks** (uint32\_t index, uint32\_t num) const
- uint32\_t **addTexel** ([Shader::Mask](#) mask)
- texel parameters*
- uint32\_t **getNumTexels** () const
- [Traversal](#) & **setTexelOffset** (uint32\_t offset)
- uint32\_t **getTexelOffset** () const
- [Traversal](#) & **setTexelMask** (uint32\_t index, [Shader::Mask](#) mask)
- [Shader::Mask](#) **getTexelMask** (uint32\_t index) const
- [Traversal](#) & **setTexelMasks** (uint32\_t index, uint32\_t num, [Shader::Mask](#) mask)
- [Shader::Mask](#) **getTexelMasks** (uint32\_t index, uint32\_t num) const
- uint32\_t **addTable** (TableType type, uint32\_t size, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
- table parameters*
- uint32\_t **getNumTables** () const
- [Traversal](#) & **setTableOffset** (uint32\_t offset)
- uint32\_t **getTableOffset** () const
- [Traversal](#) & **setTableType** (uint32\_t index, TableType type, uint32\_t size, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
- TableType **getTableType** (uint32\_t index) const
- uint32\_t **getTableSize** (uint32\_t index) const
- [Traversal](#) & **setTableMask** (uint32\_t index, [Shader::Mask](#) mask, BindFlags flags=BindFlagNone)
- [Shader::Mask](#) **getTableMask** (uint32\_t index) const
- [Traversal](#) & **setTableFlags** (uint32\_t index, BindFlags flags)
- BindFlags **getTableFlags** (uint32\_t index) const
- void **setRecursionDepth** (uint32\_t depth)
- recursion depth*
- uint32\_t **getRecursionDepth** () const

### 5.268.1 Detailed Description

[Traversal](#) class

## 5.269 Tellusim::uint16x8\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

### Public Types

- enum { **Size** = 8 }

### Public Member Functions

- [uint16x8\\_t](#) (const [uint32x8\\_t](#) &v)
- [uint16x8\\_t](#) (const uint16\_t \*v)
- [uint16x8\\_t](#) (const [uint32x4\\_t](#) &v0, const [uint32x4\\_t](#) &v1)
- [uint16x8\\_t](#) (uint16\_t v)
- [uint16x8\\_t](#) (uint16\_t x0, uint16\_t y0, uint16\_t z0, uint16\_t w0, uint16\_t x1, uint16\_t y1, uint16\_t z1, uint16\_t w1)
- [int16x8\\_t asi16x8](#) () const  
*cast vector data*
- [int32x4\\_t asi32x4](#) () const
- [uint32x4\\_t asu32x4](#) () const  
*cast vector data*
- [float16x8\\_t asf16x8](#) () const
- [float32x4\\_t asf32x4](#) () const
- void [set](#) (const [uint16x8\\_t](#) &v)  
*update vector data*
- void [set](#) (uint16\_t X0, uint16\_t Y0, uint16\_t Z0, uint16\_t W0, uint16\_t X1, uint16\_t Y1, uint16\_t Z1, uint16\_t W1)
- void [set](#) (const uint16\_t \*1 v)
- void [get](#) (uint16\_t \*1 v) const
- template<uint32\_t Index>  
void [set](#) (uint16\_t V)
- template<uint32\_t Index>  
uint16\_t [get](#) () const
- [uint16x8\\_t & operator\\*=](#) (uint16\_t v)  
*vector to scalar operators*
- [uint16x8\\_t & operator+=](#) (uint16\_t v)
- [uint16x8\\_t & operator-=](#) (uint16\_t v)
- [uint16x8\\_t & operator &=](#) (uint16\_t v)
- [uint16x8\\_t & operator|=](#) (uint16\_t v)
- [uint16x8\\_t & operator^=](#) (uint16\_t v)
- [uint16x8\\_t & operator<<=](#) (uint16\_t v)
- [uint16x8\\_t & operator>>=](#) (uint16\_t v)
- [uint16x8\\_t & operator\\*=](#) (const [uint16x8\\_t](#) &v)  
*vector to vector operators*
- [uint16x8\\_t & operator+=](#) (const [uint16x8\\_t](#) &v)
- [uint16x8\\_t & operator-=](#) (const [uint16x8\\_t](#) &v)
- [uint16x8\\_t & operator &=](#) (const [uint16x8\\_t](#) &v)
- [uint16x8\\_t & operator|=](#) (const [uint16x8\\_t](#) &v)
- [uint16x8\\_t & operator^=](#) (const [uint16x8\\_t](#) &v)
- [uint16x8\\_t xyzw10](#) () const  
*swizzle vector*
- [uint16x8\\_t zwx01](#) () const



- [uint16x8\\_t yxwz01](#) () const  
*swizzle vector*
- [uint32x4\\_t xyzw0](#) () const
- [uint32x4\\_t xyzw1](#) () const
- [uint16\\_t sum](#) () const  
*sum vector components*

#### Public Attributes

- union {  
  struct {  
    uint16\_t **x0**  
    uint16\_t **y0**  
    uint16\_t **z0**  
    uint16\_t **w0**  
    uint16\_t **x1**  
    uint16\_t **y1**  
    uint16\_t **z1**  
    uint16\_t **w1**  
  }  
  uint16\_t **v** [[Size](#)]  
};

#### 5.269.1 Detailed Description

Vector of eight uint16\_t components

#### 5.269.2 Constructor & Destructor Documentation

##### 5.269.2.1 uint16x8\_t()

```
Tellusim::uint16x8_t::uint16x8_t (
    const uint32x8\_t & v ) [explicit]
```

Vector of eight uint16\_t components

## 5.270 Tellusim::uint32x4\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

#### Public Types

- enum { **Size** = 4 }

## Public Member Functions

- [uint32x4\\_t](#) (const [int32x4\\_t](#) &v)
- [uint32x4\\_t](#) (const [float32x4\\_t](#) &v)
- [uint32x4\\_t](#) (const [float64x4\\_t](#) &v)
- [uint32x4\\_t](#) (const uint32\_t \*v)
- [uint32x4\\_t](#) (const uint32\_t \*v, uint32\_t w)
- [uint32x4\\_t](#) (uint32\_t v)
- [uint32x4\\_t](#) (uint32\_t x, uint32\_t y, uint32\_t z, uint32\_t w=0)
- [int16x8\\_t](#) [asi16x8](#) () const

*cast vector data*

- [int32x4\\_t](#) [asi32x4](#) () const
- [uint16x8\\_t](#) [asu16x8](#) () const
- [float16x8\\_t](#) [asf16x8](#) () const

*cast vector data*

- [float32x4\\_t](#) [asf32x4](#) () const
- void [set](#) (const [uint32x4\\_t](#) &v)

*update vector data*

- void [set](#) (uint32\_t X, uint32\_t Y, uint32\_t Z, uint32\_t W)
- void [set](#) (const uint32\_t \*1 v, uint32\_t W)
- void [set](#) (const uint32\_t \*1 v)
- void [get](#) (uint32\_t \*1 v) const
- template<uint32\_t Index>  
void [set](#) (uint32\_t V)
- template<uint32\_t Index>  
uint32\_t [get](#) () const
- template<uint32\_t Index>  
[uint32x4\\_t](#) [get4](#) () const
- [uint32x4\\_t](#) & [operator\\*=](#) (uint32\_t v)

*vector to scalar operators*

- [uint32x4\\_t](#) & [operator+=](#) (uint32\_t v)
- [uint32x4\\_t](#) & [operator-=](#) (uint32\_t v)
- [uint32x4\\_t](#) & [operator &=](#) (uint32\_t v)
- [uint32x4\\_t](#) & [operator|=](#) (uint32\_t v)
- [uint32x4\\_t](#) & [operator^=](#) (uint32\_t v)
- [uint32x4\\_t](#) & [operator<<=](#) (uint32\_t v)
- [uint32x4\\_t](#) & [operator>>=](#) (uint32\_t v)
- [uint32x4\\_t](#) & [operator\\*=](#) (const [uint32x4\\_t](#) &v)

*vector to vector operators*

- [uint32x4\\_t](#) & [operator+=](#) (const [uint32x4\\_t](#) &v)
- [uint32x4\\_t](#) & [operator-=](#) (const [uint32x4\\_t](#) &v)
- [uint32x4\\_t](#) & [operator &=](#) (const [uint32x4\\_t](#) &v)
- [uint32x4\\_t](#) & [operator|=](#) (const [uint32x4\\_t](#) &v)
- [uint32x4\\_t](#) & [operator^=](#) (const [uint32x4\\_t](#) &v)
- [uint32x4\\_t](#) [zwxxy](#) () const

*swizzle vector*

- [uint32x4\\_t](#) [yxwz](#) () const
- uint32\_t [sum](#) () const

*sum vector components*

## Public Attributes

- ```
union {  
    struct {  
        uint32_t x  
        uint32_t y  
        uint32_t z  
        uint32_t w  
    }  
    uint32_t v [Size]  
};
```

## 5.270.1 Detailed Description

Vector of four uint32\_t components

## 5.270.2 Constructor &amp; Destructor Documentation

## 5.270.2.1 uint32x4\_t()

```
Tellusim::uint32x4_t::uint32x4_t (  
    const int32x4_t & v ) [explicit]
```

Vector of four uint32\_t components

## 5.271 Tellusim::uint32x8\_t Struct Reference

```
#include <math/TellusimSimd.h>
```

## Public Types

- enum { **Size** = 8 }

## Public Member Functions

- [uint32x8\\_t](#) (const [int32x8\\_t](#) &v)
- [uint32x8\\_t](#) (const [float32x8\\_t](#) &v)
- [uint32x8\\_t](#) (const [float64x8\\_t](#) &v)
- [uint32x8\\_t](#) (const uint32\_t \*v)
- [uint32x8\\_t](#) (uint32\_t v)
- [uint32x8\\_t](#) (const [uint16x8\\_t](#) &v)
- [uint32x8\\_t](#) (const [uint32x4\\_t](#) &v0, const [uint32x4\\_t](#) &v1)
- [uint32x8\\_t](#) (uint32\_t x0, uint32\_t y0, uint32\_t z0, uint32\_t w0, uint32\_t x1, uint32\_t y1, uint32\_t z1, uint32\_t w1)
- [int32x8\\_t](#) [asi32x8](#) () const  
*cast vector data*
- [float32x8\\_t](#) [asf32x8](#) () const
- void [set](#) (const [uint32x8\\_t](#) &v)  
*update vector data*
- void [set](#) (uint32\_t X0, uint32\_t Y0, uint32\_t Z0, uint32\_t W0, uint32\_t X1, uint32\_t Y1, uint32\_t Z1, uint32\_t W1)
- void [set](#) (const uint32\_t \*1 v)
- void [get](#) (uint32\_t \*1 v) const
- template<uint32\_t Index>  
void [set](#) (uint32\_t V)
- template<uint32\_t Index>  
uint32\_t [get](#) () const
- template<uint32\_t Index>  
[uint32x8\\_t](#) [get8](#) () const
- [uint32x8\\_t](#) & [operator\\*=](#) (uint32\_t v)  
*vector to scalar operators*
- [uint32x8\\_t](#) & [operator+=](#) (uint32\_t v)
- [uint32x8\\_t](#) & [operator-=](#) (uint32\_t v)
- [uint32x8\\_t](#) & [operator &=](#) (uint32\_t v)
- [uint32x8\\_t](#) & [operator|=](#) (uint32\_t v)
- [uint32x8\\_t](#) & [operator^=](#) (uint32\_t v)
- [uint32x8\\_t](#) & [operator<<=](#) (uint32\_t v)
- [uint32x8\\_t](#) & [operator>>=](#) (uint32\_t v)
- [uint32x8\\_t](#) & [operator\\*=](#) (const [uint32x8\\_t](#) &v)  
*vector to vector operators*
- [uint32x8\\_t](#) & [operator+=](#) (const [uint32x8\\_t](#) &v)
- [uint32x8\\_t](#) & [operator-=](#) (const [uint32x8\\_t](#) &v)
- [uint32x8\\_t](#) & [operator &=](#) (const [uint32x8\\_t](#) &v)
- [uint32x8\\_t](#) & [operator|=](#) (const [uint32x8\\_t](#) &v)
- [uint32x8\\_t](#) & [operator^=](#) (const [uint32x8\\_t](#) &v)
- [uint32x8\\_t](#) [xyzw10](#) () const  
*swizzle vector*
- [uint32x8\\_t](#) [zwxy01](#) () const
- [uint32x8\\_t](#) [yxwz01](#) () const
- [uint32x4\\_t](#) [xyzw0](#) () const
- [uint32x4\\_t](#) [xyzw1](#) () const
- uint32\_t [sum](#) () const  
*sum vector components*

## Public Attributes

```

•
union {
    struct {
        uint32_t x0
        uint32_t y0
        uint32_t z0
        uint32_t w0
        uint32_t x1
        uint32_t y1
        uint32_t z1
        uint32_t w1
    }
    uint32_t v [Size]
};

```

## 5.271.1 Detailed Description

Vector of eight uint32\_t components

## 5.271.2 Constructor &amp; Destructor Documentation

## 5.271.2.1 uint32x8\_t()

```

Tellusim::uint32x8_t::uint32x8_t (
    const int32x8_t & v ) [explicit]

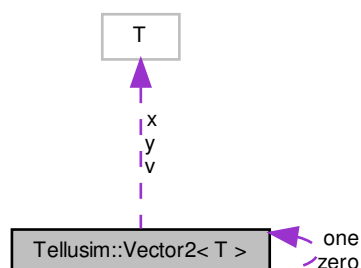
```

Vector of eight uint32\_t components

## 5.272 Tellusim::Vector2&lt; T &gt; Struct Template Reference

```
#include <math/TellusimVector.h>
```

Collaboration diagram for Tellusim::Vector2< T >:



## Public Types

- enum { **Size** = 2 }

## Public Member Functions

- **Vector2** (const [Vector2](#) &v)
- **Vector2** (const Type &x, const Type &y)
- template<class CType >  
**Vector2** (const [Vector2](#)< CType > &v)
- template<class CType >  
**Vector2** (const [Vector3](#)< CType > &v)
- template<class CType >  
**Vector2** (const [Vector4](#)< CType > &v)
- **Vector2** (const Type \*1 v)
- **Vector2** (const Type &v)
- void [set](#) (const Type &v)
- update vector data*
- void **set** (const Type &X, const Type &Y)
- void **set** (const [Vector3](#)< Type > &v)
- void **set** (const [Vector4](#)< Type > &v)
- void **set** (const Type \*1 v)
- void **get** (Type \*1 v) const
- [Vector2](#) & [operator](#)\*= (const Type &v)
- vector to scalar operators*
- [Vector2](#) & [operator](#)/= (const Type &v)
- [Vector2](#) & [operator](#)%= (const Type &v)
- [Vector2](#) & [operator](#)+= (const Type &v)
- [Vector2](#) & [operator](#)-= (const Type &v)
- [Vector2](#) & [operator](#) &= (const Type &v)
- [Vector2](#) & [operator](#)|= (const Type &v)
- [Vector2](#) & [operator](#)^= (const Type &v)
- [Vector2](#) & [operator](#)<<= (const Type &v)
- [Vector2](#) & [operator](#)>>= (const Type &v)
- [Vector2](#) & [operator](#)\*= (const [Vector2](#) &v)
- vector to vector operators*
- [Vector2](#) & [operator](#)/= (const [Vector2](#) &v)
- [Vector2](#) & [operator](#)%= (const [Vector2](#) &v)
- [Vector2](#) & [operator](#)+= (const [Vector2](#) &v)
- [Vector2](#) & [operator](#)-= (const [Vector2](#) &v)
- [Vector2](#) & [operator](#) &= (const [Vector2](#) &v)
- [Vector2](#) & [operator](#)|= (const [Vector2](#) &v)
- [Vector2](#) & [operator](#)^= (const [Vector2](#) &v)
- const Type & [operator](#)[] (uint32\_t index) const
- vector data*
- Type & [operator](#)[] (uint32\_t index)
- Type [cartesian](#) () const
- homogeneous transform*

## Public Attributes

- ```
union {  
    struct {  
        Type x  
        Type y  
    }  
    Type v [Size]  
};
```

## Static Public Attributes

- static const [Vector2 zero](#)  
*default vectors*
- static const [Vector2 one](#)

## 5.272.1 Detailed Description

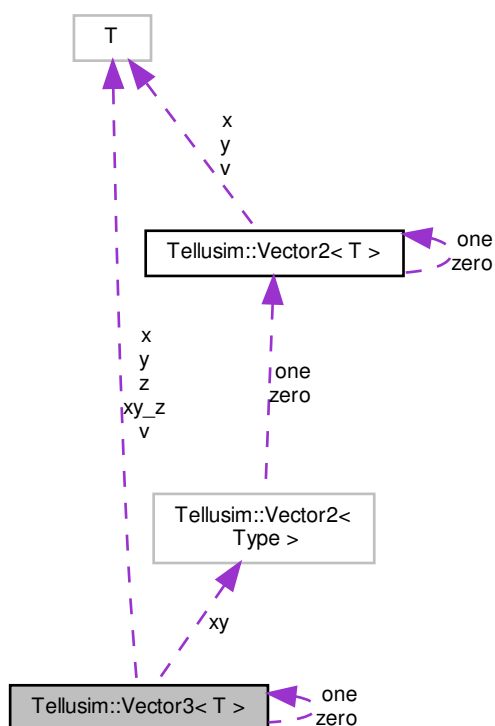
```
template<class T>  
struct Tellusim::Vector2< T >
```

[Vector2](#) class

## 5.273 Tellusim::Vector3&lt; T &gt; Struct Template Reference

```
#include <math/TellusimVector.h>
```

Collaboration diagram for Tellusim::Vector3< T >:



### Public Types

- enum { **Size** = 3 }

### Public Member Functions

- **Vector3** (const [Vector3](#) &v)
- **Vector3** (const Type &x, const Type &y, const Type &z)
- template<class CType >  
**Vector3** (const [Vector2](#)< CType > &v, const CType &z)
- template<class CType >  
**Vector3** (const [Vector3](#)< CType > &v)
- template<class CType >  
**Vector3** (const [Vector4](#)< CType > &v)
- **Vector3** (const Type \*1 v)
- **Vector3** (const Type &v)
- void [set](#) (const Type &v)  
*update vector data*
- void **set** (const Type &X, const Type &Y, const Type &Z)
- void **set** (const [Vector2](#)< Type > &v, const Type &Z)
- void **set** (const [Vector4](#)< Type > &v)
- void **set** (const Type \*1 v)



- void **get** (Type \*1 v) const
- **Vector3** & **operator\*=** (const Type &v)  
*vector to scalar operators*
- **Vector3** & **operator/=** (const Type &v)
- **Vector3** & **operator%=** (const Type &v)
- **Vector3** & **operator+=** (const Type &v)
- **Vector3** & **operator-=** (const Type &v)
- **Vector3** & **operator &=** (const Type &v)
- **Vector3** & **operator|=** (const Type &v)
- **Vector3** & **operator^=** (const Type &v)
- **Vector3** & **operator<<=** (const Type &v)
- **Vector3** & **operator>>=** (const Type &v)
- **Vector3** & **operator\*=** (const **Vector3** &v)  
*vector to vector operators*
- **Vector3** & **operator/=** (const **Vector3** &v)
- **Vector3** & **operator%=** (const **Vector3** &v)
- **Vector3** & **operator+=** (const **Vector3** &v)
- **Vector3** & **operator-=** (const **Vector3** &v)
- **Vector3** & **operator &=** (const **Vector3** &v)
- **Vector3** & **operator|=** (const **Vector3** &v)
- **Vector3** & **operator^=** (const **Vector3** &v)
- const Type & **operator[]** (uint32\_t index) const  
*vector data*
- Type & **operator[]** (uint32\_t index)
- **Vector2**< Type > **cartesian** () const  
*homogeneous transform*

#### Public Attributes

- union {  
  struct {  
    Type **x**  
    Type **y**  
    Type **z**  
  }  
  struct {  
    **Vector2**< Type > **xy**  
    Type **xy\_z**  
  }  
  Type **v** [**Size**]  
};

#### Static Public Attributes

- static const **Vector3** **zero**  
*default vectors*
- static const **Vector3** **one**

### 5.273.1 Detailed Description

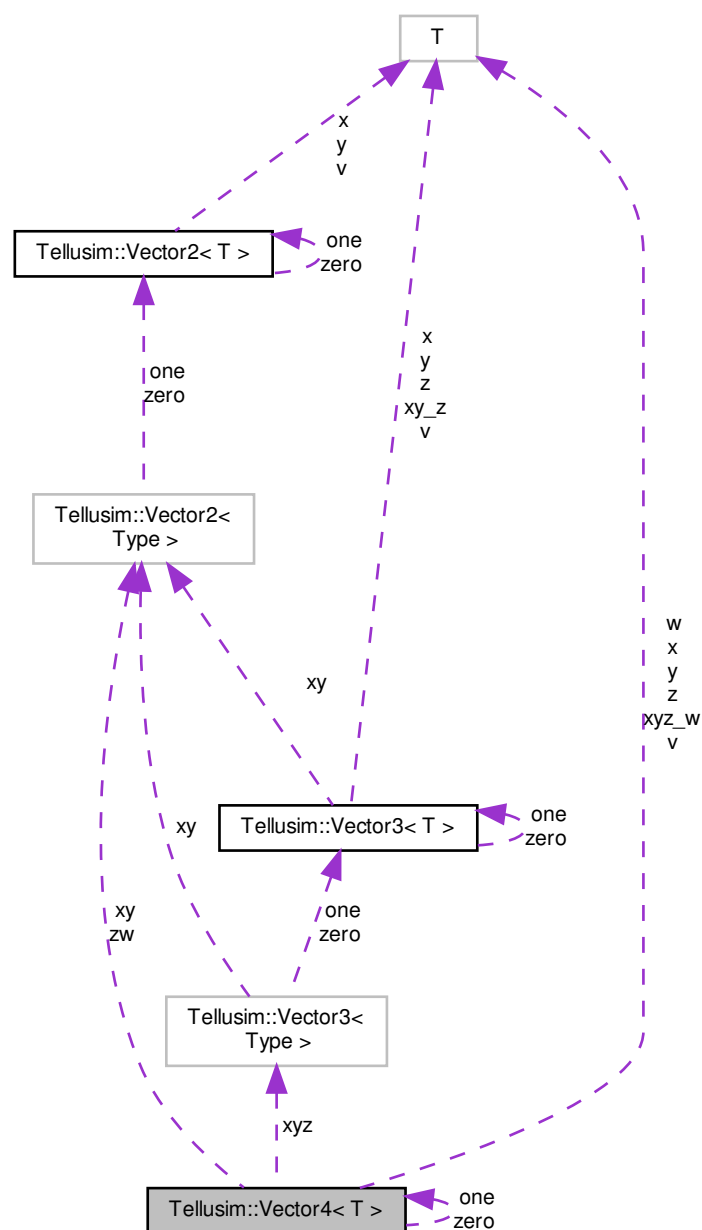
```
template<class T>
struct Tellusim::Vector3< T >
```

[Vector3](#) class

### 5.274 Tellusim::Vector4< T > Struct Template Reference

```
#include <math/TellusimVector.h>
```

Collaboration diagram for Tellusim::Vector4< T >:



#### Public Types

- enum { **Size** = 4 }

#### Public Member Functions

- Vector4** (const [Vector4](#) &*v*)

- **Vector4** (const Type &x, const Type &y, const Type &z, const Type &w)
- template<class CType >  
**Vector4** (const **Vector2**< CType > &v0, const **Vector2**< CType > &v1)
- template<class CType >  
**Vector4** (const **Vector2**< CType > &v, const CType &z, const CType &w)
- template<class CType >  
**Vector4** (const **Vector3**< CType > &v, const CType &w)
- template<class CType >  
**Vector4** (const **Vector4**< CType > &v)
- **Vector4** (const Type \*1 v)
- **Vector4** (const Type &v)
- void **set** (const Type &v)  
*update vector data*
- void **set** (const Type &X, const Type &Y, const Type &Z, const Type &W)
- void **set** (const **Vector2**< Type > &v, const Type &Z, const Type &W)
- void **set** (const **Vector3**< Type > &v, const Type &W)
- void **set** (const Type \*1 v)
- void **get** (Type \*1 v) const
- **Vector4** & **operator\*=** (const Type &v)  
*vector to scalar operators*
- **Vector4** & **operator/=** (const Type &v)
- **Vector4** & **operator%=** (const Type &v)
- **Vector4** & **operator+=** (const Type &v)
- **Vector4** & **operator-=** (const Type &v)
- **Vector4** & **operator &=** (const Type &v)
- **Vector4** & **operator|=** (const Type &v)
- **Vector4** & **operator^=** (const Type &v)
- **Vector4** & **operator<<=** (const Type &v)
- **Vector4** & **operator>>=** (const Type &v)
- **Vector4** & **operator\*=** (const **Vector4** &v)  
*vector to vector operators*
- **Vector4** & **operator/=** (const **Vector4** &v)
- **Vector4** & **operator%=** (const **Vector4** &v)
- **Vector4** & **operator+=** (const **Vector4** &v)
- **Vector4** & **operator-=** (const **Vector4** &v)
- **Vector4** & **operator &=** (const **Vector4** &v)
- **Vector4** & **operator|=** (const **Vector4** &v)
- **Vector4** & **operator^=** (const **Vector4** &v)
- const Type & **operator[]** (uint32\_t index) const  
*vector data*
- Type & **operator[]** (uint32\_t index)
- **Vector3**< Type > **cartesian** () const  
*homogeneous transform*

## Public Attributes

•

```

union {
    struct {
        Type x
        Type y
        Type z
        Type w
    }
    struct {
        Vector2< Type > xy
        Vector2< Type > zw
    }
    struct {
        Vector3< Type > xyz
        Type xyz_w
    }
    Type v [Size]
};

```

#### Static Public Attributes

- static const [Vector4](#) **zero**  
*default vectors*
- static const [Vector4](#) **one**

#### 5.274.1 Detailed Description

```

template<class T>
struct Tellusim::Vector4< T >

```

[Vector4](#) class

## 5.275 Tellusim::VectorN< Type, N > Struct Template Reference

```
#include <math/TellusimNumerical.h>
```

#### Public Types

- enum { **Size** = N }

#### Public Member Functions

- **VectorN** (const [VectorN](#) &vector)
- **VectorN** (const Type &value)
- **VectorN** (const Type \*vector)
- **VectorN** (const InitializerList< Type > &list)
- template<class CType >  
**VectorN** (const [VectorN](#)< CType, N > &vector)
- void [set](#) (const Type &value)  
*update vector data*

- void **set** (const Type \*1 vector)
- void **set** (const [VectorN](#) &vector)
- void **set** (const InitializerList< Type > &list)
- void **get** (Type \*1 vector)
- [VectorN](#) & **operator\*=** (const Type &value)  
*vector to scalar operators*
- [VectorN](#) & **operator/=** (const Type &value)
- [VectorN](#) & **operator+=** (const Type &value)
- [VectorN](#) & **operator-=** (const Type &value)
- [VectorN](#) & **operator=** (const [VectorN](#) &vector)  
*vector to vector operators*
- [VectorN](#) & **operator\*=** (const [VectorN](#) &vector)
- [VectorN](#) & **operator/=** (const [VectorN](#) &vector)
- [VectorN](#) & **operator+=** (const [VectorN](#) &vector)
- [VectorN](#) & **operator-=** (const [VectorN](#) &vector)
- const Type & **operator[]** (uint32\_t index) const  
*vector data*
- Type & **operator[]** (uint32\_t index)

#### Public Attributes

- Type **v** [[Size](#)]

#### 5.275.1 Detailed Description

```
template<class Type, uint32_t N>
struct Tellusim::VectorN< Type, N >
```

[VectorN](#) class

#### 5.276 Tellusim::Viewport Struct Reference

```
#include <TellusimTypes.h>
```

#### Public Member Functions

- **Viewport** (float32\_t width, float32\_t height)
- **Viewport** (float32\_t x, float32\_t y, float32\_t width, float32\_t height)
- **Viewport** (float32\_t x, float32\_t y, float32\_t width, float32\_t height, float32\_t znear, float32\_t zfar)
- float32\_t **getLeft** () const
- float32\_t **getBottom** () const
- float32\_t **getRight** () const
- float32\_t **getTop** () const

## Public Attributes

- float32\_t **x** = 0.0f
- float32\_t **y** = 0.0f
- float32\_t **width** = 1.0f
- float32\_t **height** = 1.0f
- float32\_t **znear** = 0.0f
- float32\_t **zfar** = 1.0f

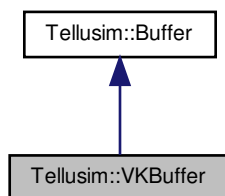
## 5.276.1 Detailed Description

[Viewport](#)

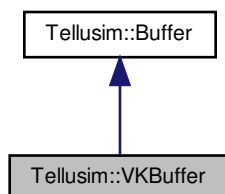
## 5.277 Tellusim::VKBuffer Class Reference

```
#include <platform/TellusimBuffer.h>
```

Inheritance diagram for Tellusim::VKBuffer:



Collaboration diagram for Tellusim::VKBuffer:



## Public Member Functions

- bool **create** ([Flags](#) flags, size\_t size, VkBuffer buffer, uint32\_t access)  
*create external buffer*
- VkBuffer **getVkBuffer** () const
- VkBufferView **getBufferView** () const
- uint64\_t **getBufferAddress** () const
- void **setBufferAccess** (uint32\_t access)
- uint32\_t **getBufferAccess** () const
- void \* **getSharedPtr** () const
- void \* **getInteropHandle** () const

## Additional Inherited Members

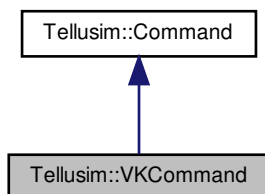
## 5.277.1 Detailed Description

[VkBuffer](#)

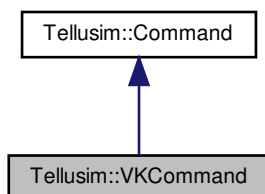
## 5.278 Tellusim::VKCommand Class Reference

```
#include <platform/TellusimCommand.h>
```

Inheritance diagram for Tellusim::VKCommand:



Collaboration diagram for Tellusim::VKCommand:





## Public Member Functions

- VkCommandBuffer [getVKCommand](#) () const  
*command context*
- VkDescriptorSet [getSamplerDescriptor](#) () const  
*command descriptors*
- VkDescriptorSet **getImageDescriptor** () const
- VkDescriptorSet **getBufferDescriptor** () const
- VkDescriptorSet **getTracingDescriptor** () const
- VkDescriptorSet **getTexelDescriptor** () const
- void [update](#) ()  
*update resources*

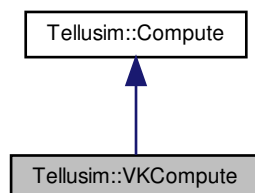
## 5.278.1 Detailed Description

[VKCommand](#)

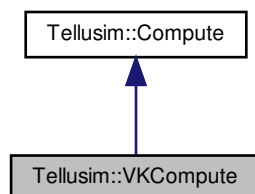
## 5.279 Tellusim::VKCompute Class Reference

```
#include <platform/TellusimCompute.h>
```

Inheritance diagram for Tellusim::VKCompute:



Collaboration diagram for Tellusim::VKCompute:



## Public Member Functions

- VkCommandBuffer [getCommand](#) () const  
*command context*
- VkDescriptorSet [getSamplerDescriptor](#) () const  
*compute descriptors*
- VkDescriptorSet **getImageDescriptor** () const
- VkDescriptorSet **getBufferDescriptor** () const
- VkDescriptorSet **getTracingDescriptor** () const
- VkDescriptorSet **getTexelDescriptor** () const
- void [update](#) ()  
*update resources*

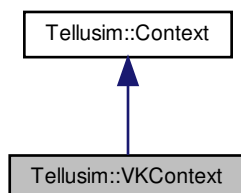
## 5.279.1 Detailed Description

[VKCompute](#)

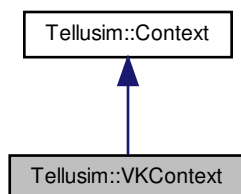
## 5.280 Tellusim::VKContext Class Reference

```
#include <platform/TellusimContext.h>
```

Inheritance diagram for Tellusim::VKContext:



Collaboration diagram for Tellusim::VKContext:



## Public Member Functions

- bool [create](#) (VkInstance instance, PFN\_vkGetInstanceProcAddr func, VkPhysicalDevice adapter, VkDevice device, uint32\_t family, uint32\_t index)  
*create context*
- VkInstance [getInstance](#) () const  
*current device*
- VkPhysicalDevice [getAdapter](#) () const
- VkDevice [getDevice](#) () const
- VkQueue [getQueue](#) () const  
*current context*
- VkCommandBuffer [getCommand](#) () const
- uint32\_t [getFamily](#) () const
- uint32\_t [getNumQueues](#) ()  
*device queues*
- uint32\_t [getQueueFlags](#) (uint32\_t index)
- uint32\_t [getQueueFamily](#) (uint32\_t index)

## Static Public Member Functions

- static void [addContextExtension](#) (const char \*name)  
*additional extensions*
- static void [addAdapterExtension](#) (const char \*name)
- static void [addAdapterFeatures](#) (void \*features)
- static PFN\_vkGetInstanceProcAddr [getInstanceProcAddress](#) ()  
*get proc address functions*
- static PFN\_vkGetDeviceProcAddr [getDeviceProcAddress](#) ()
- static void \* [getProcAddress](#) (const char \*name)  
*Vulkan functions.*
- static bool [error](#) (uint32\_t result)  
*check Vulkan errors*

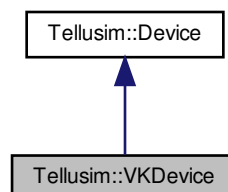
## 5.280.1 Detailed Description

[VKContext](#)

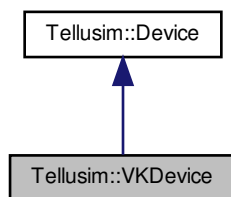
## 5.281 Tellusim::VKDevice Class Reference

```
#include <platform/TellusimDevice.h>
```

Inheritance diagram for Tellusim::VKDevice:



Collaboration diagram for Tellusim::VKDevice:



### Public Member Functions

- **VKDevice** ([Context](#) &context)
- **VKDevice** ([Surface](#) &surface)
- **VKDevice** ([Window](#) &window)
- void **setBufferAccess** ([Buffer](#) &buffer, uint32\_t access)  
*buffer access*
- void **setTextureLayout** ([Texture](#) &texture, uint32\_t layout)  
*texture layout*
- bool **waitVKFence** (void \*fence, uint64\_t timeout, bool reset) const  
*fence synchronization*
- bool **signalVKFence** (void \*fence) const
- void **waitSemaphore** (void \*semaphore, uint32\_t mask) const  
*semaphore synchronization*
- void **signalSemaphore** (void \*semaphore) const
- bool **hasMemoryType** (uint32\_t flags) const  
*memory types*
- uint32\_t **getMemoryIndex** (uint32\_t types, uint32\_t flags) const
- VkInstance **getInstance** () const  
*command context*
- VkPhysicalDevice **getAdapter** () const
- VkDevice **getVKDevice** () const
- VkQueue **getQueue** () const
- VkCommandBuffer **getCommand** () const
- uint32\_t **getFamily** () const

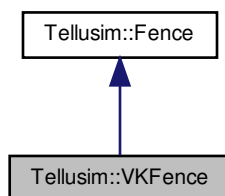
#### 5.281.1 Detailed Description

#### [VKDevice](#)

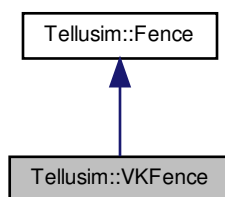
## 5.282 Tellusim::VKFence Class Reference

```
#include <platform/TellusimFence.h>
```

Inheritance diagram for Tellusim::VKFence:



Collaboration diagram for Tellusim::VKFence:



### Public Member Functions

- VkFence **getVKFence** () const
- VkSemaphore **getSemaphore** () const
- void \* **getSharedHandle** () const

### Additional Inherited Members

#### 5.282.1 Detailed Description

[VKFence](#)

## 5.283 Tellusim::VKTracing::VKInstance Struct Reference

tracing instance

```
#include <platform/TellusimTracing.h>
```

### Public Attributes

- float32\_t **transform** [12]
- uint32\_t **data**: 24
- uint32\_t **mask**: 8
- uint32\_t **offset**: 24
- uint32\_t **flags**: 8
- uint64\_t **address**

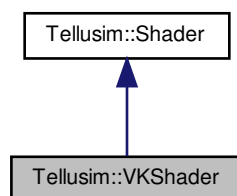
### 5.283.1 Detailed Description

tracing instance

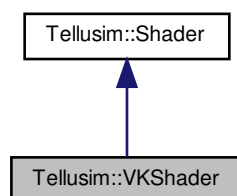
## 5.284 Tellusim::VKShader Class Reference

```
#include <platform/TellusimShader.h>
```

Inheritance diagram for Tellusim::VKShader:



Collaboration diagram for Tellusim::VKShader:



### Public Member Functions

- VkShaderModule **getModule** () const

## Additional Inherited Members

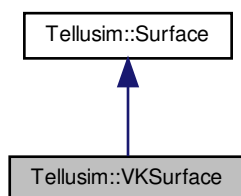
## 5.284.1 Detailed Description

[VKShader](#)

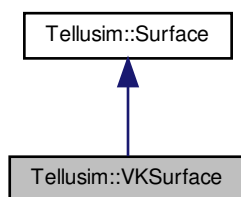
## 5.285 Tellusim::VKSurface Class Reference

```
#include <platform/TellusimSurface.h>
```

Inheritance diagram for Tellusim::VKSurface:



Collaboration diagram for Tellusim::VKSurface:



## Public Member Functions

- **VKSurface** ([VKContext](#) &context)
- VkInstance [getInstance](#) () const  
*current device*
- VkPhysicalDevice **getAdapter** () const
- VkDevice **getDevice** () const
- VkQueue **getQueue** () const
- VkCommandBuffer **getCommand** () const

- uint32\_t **getFamily** () const
- void **setColorImage** (VkImage image)
  - image handles*
- void **setDepthImage** (VkImage image)
- VkImage **getColorImage** () const
- VkImage **getDepthImage** () const
- void **setColorImageView** (VkImageView image\_view)
  - image view handles*
- void **setDepthImageView** (VkImageView image\_view)
- VkImageView **getColorImageView** () const
- VkImageView **getDepthImageView** () const
- void **setRenderPass** (VkRenderPass render\_pass)
  - framebuffer handle*
- void **setFramebuffer** (VkFramebuffer framebuffer)
- VkRenderPass **getRenderPass** () const
- VkFramebuffer **getFramebuffer** () const
- uint32\_t **getColorPixelFormat** () const
  - surface formats*
- uint32\_t **getDepthPixelFormat** () const

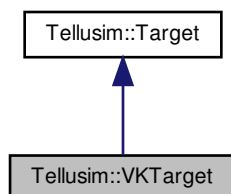
#### 5.285.1 Detailed Description

[VKSurface](#)

### 5.286 Tellusim::VKTarget Class Reference

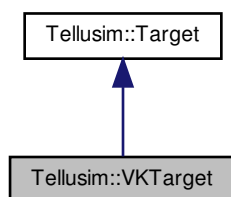
```
#include <platform/TellusimTarget.h>
```

Inheritance diagram for Tellusim::VKTarget:





Collaboration diagram for Tellusim::VKTarget:



#### Public Member Functions

- VkRenderPass **getRenderPass** () const
- VkFramebuffer **getFramebuffer** () const

#### Additional Inherited Members

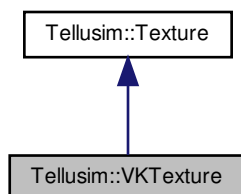
##### 5.286.1 Detailed Description

[VKTarget](#)

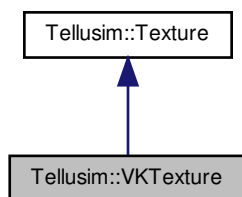
## 5.287 Tellusim::VKTexture Class Reference

```
#include <platform/TellusimTexture.h>
```

Inheritance diagram for Tellusim::VKTexture:



Collaboration diagram for Tellusim::VKTexture:



#### Public Member Functions

- `bool create` (Type type, uint32\_t format, VkImage texture, uint32\_t layout, [Flags](#) flags=DefaultFlags, Format texture\_format=FormatUnknown)  
*create external texture*
- `uint32_t getPixelFormat () const`
- `VkImage getVKTexture () const`
- `VkImageView getTextureView () const`
- `void setTextureLayout (uint32_t layout)`
- `uint32_t getTextureLayout () const`
- `void getTextureRange (void *range, const Slice &slice) const`
- `void * getSharedPtr () const`
- `void * getInteropHandle () const`

#### Additional Inherited Members

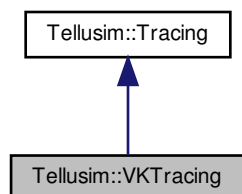
##### 5.287.1 Detailed Description

#### [VKTexture](#)

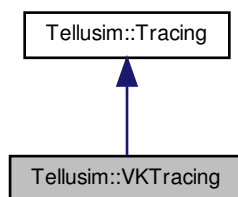
### 5.288 Tellusim::VKTracing Class Reference

```
#include <platform/TellusimTracing.h>
```

Inheritance diagram for Tellusim::VKTracing:



Collaboration diagram for Tellusim::VKTracing:



## Classes

- struct [VKInstance](#)  
*tracing instance*

## Public Member Functions

- void \* **getBuildGeometryInfo** () const
- void \* **getBuildSizeInfo** () const
- [Buffer](#) **getTracingBuffer** () const
- VkAccelerationStructureKHR **getAccelerationStructure** () const

## Additional Inherited Members

### 5.288.1 Detailed Description

## [VKTracing](#)

## 5.289 Tellusim::Window Class Reference

```
#include <platform/TellusimWindow.h>
```

## Public Types

- enum [Flags](#) {
  - FlagNone** = 0,
  - FlagTitle** = (1 << 0),
  - FlagClose** = (1 << 1),
  - FlagFrame** = (1 << 2),
  - FlagResize** = (1 << 3),
  - FlagMinimize** = (1 << 4),
  - FlagMaximize** = (1 << 5),
  - FlagTransient** = (1 << 6),
  - FlagFullscreen** = (1 << 7),
  - FlagTransparent** = (1 << 8),
  - FlagFileDropped** = (1 << 9),
  - FlagMultisample2** = (1 << 10),
  - FlagMultisample4** = (1 << 11),
  - FlagMultisample8** = (1 << 12),
  - FlagVerticalSync** = (1 << 13),
  - FlagRefreshSync** = (1 << 14),
  - FlagColorRGBAu8ns** = (1 << 15),
  - FlagColorRGBu10Au2n** = (1 << 16),
  - FlagColorRGBAf16** = (1 << 17),
  - FlagMultisample** = (FlagMultisample2 | FlagMultisample4 | FlagMultisample8),
  - DefaultFlags** = (FlagTitle | FlagClose | FlagResize | FlagMinimize | FlagMaximize),
  - NumFlags** = 18 }

*Window flags.*
- enum [Cursor](#) {
  - CursorArrow** = 0,
  - CursorInvalid**,
  - CursorLeft**,
  - CursorRight**,
  - CursorBottom**,
  - CursorTop**,
  - CursorWidth**,
  - CursorHeight**,
  - CursorMajor**,
  - CursorMinor**,
  - CursorAll**,
  - NumCursors** }

*Mouse cursors.*
- enum [Button](#) {
  - ButtonNone** = 0,
  - ButtonLeft** = (1 << 0),
  - ButtonLeft2** = (1 << 1),
  - ButtonRight** = (1 << 2),
  - ButtonRight2** = (1 << 3),
  - ButtonMiddle** = (1 << 4),
  - ButtonMiddle2** = (1 << 5),
  - ButtonAux** = (1 << 6),
  - ButtonAux2** = (1 << 7),
  - NumButtons** = 8 }

*Mouse buttons.*
- enum [Axis](#) {
  - AxisX** = 0,
  - AxisY**,
  - AxisZ**,
  - AxisW**,
  - NumAxes** }

*Mouse axes.*

- enum { **NumTouches** = 16 }

*Screen touches.*

- enum **Key** {  
**KeyNone** = 128,  
**KeyEsc**,  
**KeyTab**,  
**KeyBackspace**,  
**KeyDelete**,  
**KeyInsert**,  
**KeyReturn**,  
**KeyPause**,  
**KeyPrior**,  
**KeyNext**,  
**KeyEnd**,  
**KeyHome**,  
**KeyUp**,  
**KeyDown**,  
**KeyLeft**,  
**KeyRight**,  
**KeyNum**,  
**KeyCaps**,  
**KeyScroll**,  
**KeyShift**,  
**KeyCtrl**,  
**KeyAlt**,  
**KeyWin**,  
**KeyCmd**,  
**KeyMenu**,  
**KeyF1**,  
**KeyF2**,  
**KeyF3**,  
**KeyF4**,  
**KeyF5**,  
**KeyF6**,  
**KeyF7**,  
**KeyF8**,  
**KeyF9**,  
**KeyF10**,  
**KeyF11**,  
**KeyF12**,  
**NumKeys**,  
**KeyOption** = KeyCtrl }

*Keyboard keys.*

- using **MousePressedCallback** = Function< void(**Button** button)>  
*mouse pressed callback*
- using **MouseReleasedCallback** = Function< void(**Button** button)>  
*mouse released callback*
- using **MouseChangedCallback** = Function< void(int32\_t x, int32\_t y)>  
*mouse changed callback*
- using **MouseRotatedCallback** = Function< void(**Axis** axis, float32\_t delta)>  
*mouse rotated callback*
- using **TouchChangedCallback** = Function< void()>  
*touch changed callback*
- using **KeyboardPressedCallback** = Function< void(uint32\_t key, uint32\_t code)>  
*keyboard pressed callback*

- using [KeyboardReleasedCallback](#) = Function< void(uint32\_t key)>  
*keyboard released callback*
- using [SizeChangedCallback](#) = Function< void(uint32\_t width, uint32\_t height)>  
*size changed callback*
- using [FocusChangedCallback](#) = Function< void(bool changed)>  
*focus changed callback*
- using [CloseClickedCallback](#) = Function< void()>  
*close clicked callback*
- using [PauseChangedCallback](#) = Function< void(bool paused)>  
*pause changed callback*
- using [FileDroppedCallback](#) = Function< void(const char \*name, uint32\_t remain)>  
*file dropped callback*
- using [UpdateCallback](#) = Function< void()>  
*update callback*
- using [PresentCallback](#) = Function< void()>  
*present callback*
- using [MainLoopCallback](#) = Function< bool()>  
*window main loop callback*

#### Public Member Functions

- [Window](#) ()  
*window constructor*
- **Window** (Platform platform, uint32\_t index=Maxu32)
- **Window** ([Surface](#) &surface)
- Platform [getPlatform](#) () const  
*window platform*
- const char \* **getPlatformName** () const
- uint32\_t [getIndex](#) () const  
*window device index*
- void [setSurface](#) ([Surface](#) &surface)  
*window surface*
- [Surface](#) **getSurface** () const
- void \* [getHandle](#) () const  
*window handle*
- virtual bool [isCreated](#) () const  
*check window*
- virtual bool [create](#) (const char \*title, [Flags](#) flags=DefaultFlags)  
*create window*
- virtual bool **create** (const [String](#) &title, [Flags](#) flags=DefaultFlags)
- virtual bool **create** ([Flags](#) flags=DefaultFlags)
- virtual void **release** ()
- bool [clear](#) (const [Color](#) &color)  
*clear window*
- bool [grab](#) ([Image](#) &image) const  
*grab window*
- virtual bool [render](#) ()  
*render window*
- virtual bool **present** ()
- virtual bool **finish** ()
- Format [getColorFormat](#) () const

*window format*

- Format **getDepthFormat** () const
- uint32\_t **getMultisample** () const
- bool **hasMultisample** () const
- virtual void **setFlags** (Flags flags)

*window flags*

- Flags **getFlags** () const
- bool **hasFlag** (Flags flags) const
- bool **hasFlags** (Flags flags) const
- void **setRefreshRate** (uint32\_t rate)

*window refresh rate*

- uint32\_t **getRefreshRate** () const
- virtual bool **setHidden** (bool hidden)

*hide window*

- bool **isHidden** () const
- virtual bool **setFocused** (bool focused)

*focus window*

- bool **isFocused** () const
- virtual bool **setMinimized** (bool minimized)

*minimize window*

- bool **isMinimized** () const
- virtual bool **setFullscreen** (bool fullscreen)

*fullscreen window*

- bool **isFullscreen** () const
- bool **isOccluded** () const

*occluded window*

- virtual bool **setTitle** (const char \*title)

*window title*

- virtual bool **setTitle** (const String &title)
- String **getTitle** () const
- virtual bool **setIcon** (const Image &image)

*window icon image*

- Image **getIcon** () const
- virtual bool **setGeometry** (int32\_t x, int32\_t y, uint32\_t width, uint32\_t height, bool force=false)

*window geometry*

- virtual bool **setPosition** (int32\_t x, int32\_t y, bool force=false)
- int32\_t **getPositionX** (bool title=false) const
- int32\_t **getPositionY** (bool title=false) const
- virtual bool **setSize** (uint32\_t width, uint32\_t height, bool force=false)
- uint32\_t **getWidth** () const
- uint32\_t **getHeight** () const
- float32\_t **getScale** () const
- uint32\_t **getDpiX** () const
- uint32\_t **getDpiY** () const
- virtual bool **setMouse** (int32\_t x, int32\_t y, bool force=false)

*mouse position*

- int32\_t **getMouseX** () const
- int32\_t **getMouseY** () const
- bool **setMouseDelta** (int32\_t dx, int32\_t dy)
- int32\_t **getMouseDX** () const
- int32\_t **getMouseDY** () const
- virtual bool **setMouseHidden** (bool hidden, bool force=false)

*mouse hidden flag*

- bool **isMouseHidden** () const
- virtual bool **setMouseClipped** (bool clipped, bool force=false)  
*mouse clipped flag*
- bool **isMouseClipped** () const
- bool **isMouseInside** () const
- virtual bool **setMouseCursor** (Cursor cursor, bool force=false)  
*mouse cursor*
- Cursor **getMouseCursor** () const
- bool **setMouseButtons** (Button buttons)  
*mouse buttons*
- Button **getMouseButtons** () const
- bool **setMouseButton** (Button button, bool value)
- bool **getMouseButton** (Button button, bool clear=false) const
- void **releaseMouseButtons** (Button buttons)
- Button **clearMouseButtons** ()
- bool **setMouseAxis** (Axis axis, float32\_t value)  
*mouse axes*
- float32\_t **getMouseAxis** (Axis axis) const
- float32\_t **clearMouseAxis** (Axis axis)
- void **setMousePressedCallback** (const MousePressedCallback &func)
- MousePressedCallback **getMousePressedCallback** () const
- void **setMouseReleasedCallback** (const MouseReleasedCallback &func)
- MouseReleasedCallback **getMouseReleasedCallback** () const
- void **setMouseChangedCallback** (const MouseChangedCallback &func)
- MouseChangedCallback **getMouseChangedCallback** () const
- void **setMouseRotatedCallback** (const MouseRotatedCallback &func)
- MouseRotatedCallback **getMouseRotatedCallback** () const
- uint32\_t **getNumTouches** () const  
*touches*
- uint32\_t **addTouch** (int32\_t x, int32\_t y)
- int32\_t **getTouchX** (uint32\_t touch) const
- int32\_t **getTouchY** (uint32\_t touch) const
- uint32\_t **findTouch** (int32\_t x, int32\_t y) const
- void **clearTouches** ()
- void **setTouchChangedCallback** (const TouchChangedCallback &func)
- TouchChangedCallback **getTouchChangedCallback** () const
- void **setKeyboardKey** (uint32\_t key, bool value)  
*keyboard keys*
- bool **getKeyboardKey** (uint32\_t key, bool clear=false) const
- void **setKeyboardPressedCallback** (const KeyboardPressedCallback &func)
- KeyboardPressedCallback **getKeyboardPressedCallback** () const
- void **setKeyboardReleasedCallback** (const KeyboardReleasedCallback &func)
- KeyboardReleasedCallback **getKeyboardReleasedCallback** () const
- void **setSizeChangedCallback** (const SizeChangedCallback &func)
- SizeChangedCallback **getSizeChangedCallback** () const
- void **setFocusChangedCallback** (const FocusChangedCallback &func)
- FocusChangedCallback **getFocusChangedCallback** () const
- void **setCloseClickedCallback** (const CloseClickedCallback &func)
- CloseClickedCallback **getCloseClickedCallback** () const
- void **setPauseChangedCallback** (const PauseChangedCallback &func)
- PauseChangedCallback **getPauseChangedCallback** () const
- void **setFileDroppedCallback** (const FileDroppedCallback &func)
- FileDroppedCallback **getFileDroppedCallback** () const
- void **setUpdateCallback** (const UpdateCallback &func)



- [UpdateCallback](#) **getUpdateCallback** () const
- void **setPresentCallback** (const [PresentCallback](#) &func)
- [PresentCallback](#) **getPresentCallback** () const
- [MainLoopCallback](#) **getMainLoopCallback** () const
- virtual bool **run** (const [MainLoopCallback](#) &func)
- virtual bool **isRunning** () const
- virtual void **stop** ()
- bool **setCopyText** (const char \*text)  
*window copy/paste buffer*
- bool **setCopyText** (const [String](#) &text)
- [String](#) **getPasteText** () const

#### Static Public Member Functions

- static uint32\_t **getNumWindows** ()  
*all windows*
- static [Window](#) **getWindow** (uint32\_t index)
- static void **update** (bool wait=false)  
*update windows*

#### 5.289.1 Detailed Description

#### [Window](#)

## 5.290 Tellusim::Xml Class Reference

```
#include <format/TellusimXml.h>
```

#### Public Member Functions

- **Xml** (const char \*name, const char \*attributes=nullptr)
- **Xml** (const [String](#) &name, const char \*attributes=nullptr)
- **Xml** ([Xml](#) \*parent, const char \*name, const char \*attributes=nullptr)
- **Xml** ([Xml](#) \*parent, const [String](#) &name, const char \*attributes=nullptr)
- void **clear** ()  
*clear xml*
- bool **create** (const char \*str, size\_t size=0, bool owner=false)  
*create xml*
- bool **create** (const [String](#) &str, size\_t size=0, bool owner=false)
- bool **load** (const char \*name)  
*load xml*
- bool **load** (const [String](#) &name)
- bool **load** ([Stream](#) &stream)
- bool **save** (const char \*name, bool compact=false) const  
*save xml*
- bool **save** (const [String](#) &name, bool compact=false) const
- bool **save** ([Stream](#) &stream, bool compact=false) const
- const [Xml](#) **getRoot** () const  
*xml root*

- [Xml](#) **getRoot** ()
- uint32\_t **setParent** ([Xml](#) &parent, bool check=true)
  - xml parent*
- const [Xml](#) **getParent** () const
- [Xml](#) **getParent** ()
- [Xml](#) **addChild** (const char \*name, bool check=true)
  - xml children*
- uint32\_t **addChild** ([Xml](#) &child, bool check=true)
- bool **removeChild** ([Xml](#) &child)
- void **releaseChildren** ()
- uint32\_t **findChild** (const char \*name) const
- bool **isChild** (const char \*name) const
- const [Xml](#) **getChild** (const char \*name) const
- [Xml](#) **getChild** (const char \*name)
- uint32\_t **getNumChildren** () const
- const Array< [Xml](#) > **getChildren** () const
- Array< [Xml](#) > **getChildren** ()
- const [Xml](#) **getChild** (uint32\_t index) const
- [Xml](#) **getChild** (uint32\_t index)
- [String](#) **getPathName** () const
  - xml path name*
- void **setName** (const char \*name)
  - xml name*
- void **setName** (const [String](#) &name)
- [String](#) **getName** () const
- void **setData** (bool value)
  - xml data*
- void **setData** (const char \*value)
- void **setData** (const [String](#) &value)
- void **setData** (int32\_t value, uint32\_t radix=10)
- void **setData** (uint32\_t value, uint32\_t radix=10)
- void **setData** (uint64\_t value, uint32\_t radix=10)
- void **setData** (float32\_t value, uint32\_t digits=6, bool compact=true, bool exponent=true)
- void **setData** (float64\_t value, uint32\_t digits=12, bool compact=true, bool exponent=true)
- template<class Type >
  - [Xml](#) **setData** (const char \*name, Type value)
- [String](#) **getData** () const
- bool **getDataBool** () const
- int32\_t **getDatai32** (uint32\_t radix=10) const
- uint32\_t **getDatau32** (uint32\_t radix=10) const
- uint64\_t **getDatau64** (uint32\_t radix=10) const
- float32\_t **getDataf32** () const
- float64\_t **getDataf64** () const
- [String](#) **getData** (const char \*name, const [String](#) &value=[String::null](#)) const
- bool **getData** (const char \*name, bool value) const
- int32\_t **getData** (const char \*name, int32\_t value, uint32\_t radix=10) const
- uint32\_t **getData** (const char \*name, uint32\_t value, uint32\_t radix=10) const
- uint64\_t **getData** (const char \*name, uint64\_t value, uint32\_t radix=10) const
- float32\_t **getData** (const char \*name, float32\_t value) const
- float64\_t **getData** (const char \*name, float64\_t value) const
- void **setData** (const char \*\*values, uint32\_t size, uint32\_t wrap=Maxu32)
  - xml array data*
- void **setData** (const [String](#) \*values, uint32\_t size, uint32\_t wrap=Maxu32)
- void **setData** (const int32\_t \*values, uint32\_t size, uint32\_t radix=10, uint32\_t wrap=Maxu32)

- void **setData** (const uint32\_t \*values, uint32\_t size, uint32\_t radix=10, uint32\_t wrap=Maxu32)
- void **setData** (const float32\_t \*values, uint32\_t size, uint32\_t digits=6, bool compact=true, bool exponent=true, uint32\_t wrap=Maxu32)
- void **setData** (const float64\_t \*values, uint32\_t size, uint32\_t digits=12, bool compact=true, bool exponent=true, uint32\_t wrap=Maxu32)
- template<class Type >  
**Xml setData** (const char \*name, Type \*values, uint32\_t size)
- template<class Type >  
void **setData** (const Array< Type > &values)
- template<class Type >  
void **setData** (const char \*name, const Array< Type > &values)
- uint32\_t **getData** (String \*values, uint32\_t size) const
- uint32\_t **getData** (int32\_t \*values, uint32\_t size, uint32\_t radix=10) const
- uint32\_t **getData** (uint32\_t \*values, uint32\_t size, uint32\_t radix=10) const
- uint32\_t **getData** (float32\_t \*values, uint32\_t size) const
- uint32\_t **getData** (float64\_t \*values, uint32\_t size) const
- template<class Type >  
uint32\_t **getData** (const char \*name, Type \*values, uint32\_t size) const
- template<class Type >  
uint32\_t **getData** (Array< Type > &values) const
- template<class Type >  
uint32\_t **getData** (const char \*name, Array< Type > &values) const
- uint32\_t **addAttribute** (const char \*name)  
*xml attributes*
- bool **removeAttribute** (const char \*name)
- uint32\_t **findAttribute** (const char \*name) const
- bool **isAttribute** (const char \*name) const
- void **removeAttributes** ()
- uint32\_t **getNumAttributes** () const
- String **getAttributeName** (uint32\_t index) const
- bool **setAttributes** (const char \*str)
- void **setAttribute** (uint32\_t index, bool value)
- void **setAttribute** (uint32\_t index, const char \*value)
- void **setAttribute** (uint32\_t index, const String &value)
- void **setAttribute** (uint32\_t index, int32\_t value, uint32\_t radix=10)
- void **setAttribute** (uint32\_t index, uint32\_t value, uint32\_t radix=10)
- void **setAttribute** (uint32\_t index, uint64\_t value, uint32\_t radix=10)
- void **setAttribute** (uint32\_t index, float32\_t value, uint32\_t digits=6, bool compact=true, bool exponent=true)
- void **setAttribute** (uint32\_t index, float64\_t value, uint32\_t digits=12, bool compact=true, bool exponent=true)
- template<class Type >  
uint32\_t **setAttribute** (const char \*name, Type value)
- String **setAttribute** (uint32\_t index) const
- int32\_t **getAttributei32** (uint32\_t index, uint32\_t radix=10) const
- uint32\_t **getAttributeu32** (uint32\_t index, uint32\_t radix=10) const
- uint64\_t **getAttributeu64** (uint32\_t index, uint32\_t radix=10) const
- float32\_t **getAttributef32** (uint32\_t index) const
- float64\_t **getAttributef64** (uint32\_t index) const
- String **getAttribute** (const char \*name, const String &value=String::null) const
- bool **getAttribute** (const char \*name, bool value) const
- int32\_t **getAttribute** (const char \*name, int32\_t value, uint32\_t radix=10) const
- uint32\_t **getAttribute** (const char \*name, uint32\_t value, uint32\_t radix=10) const
- uint64\_t **getAttribute** (const char \*name, uint64\_t value, uint32\_t radix=10) const
- float32\_t **getAttribute** (const char \*name, float32\_t value) const
- float64\_t **getAttribute** (const char \*name, float64\_t value) const
- void **setAttribute** (uint32\_t index, const char \*\*values, uint32\_t size, const char \*delimiter=nullptr)

*xml array attributes*

- void **setAttribute** (uint32\_t index, const [String](#) \*values, uint32\_t size, const char \*delimiter=nullptr)
- void **setAttribute** (uint32\_t index, const int32\_t \*values, uint32\_t size, uint32\_t radix=10)
- void **setAttribute** (uint32\_t index, const uint32\_t \*values, uint32\_t size, uint32\_t radix=10)
- void **setAttribute** (uint32\_t index, const float32\_t \*values, uint32\_t size, uint32\_t digits=6, bool compact=true, bool exponent=true)
- void **setAttribute** (uint32\_t index, const float64\_t \*values, uint32\_t size, uint32\_t digits=12, bool compact=true, bool exponent=true)
- template<class Type >  
uint32\_t **setAttribute** (const char \*name, Type \*values, uint32\_t size)
- template<class Type >  
void **setAttribute** (uint32\_t index, const Array< Type > &values)
- template<class Type >  
void **setAttribute** (const char \*name, const Array< Type > &values)
- uint32\_t **getAttribute** (uint32\_t index, [String](#) \*values, uint32\_t size, const char \*delimiter=nullptr) const
- uint32\_t **getAttribute** (uint32\_t index, int32\_t \*values, uint32\_t size, uint32\_t radix=10) const
- uint32\_t **getAttribute** (uint32\_t index, uint32\_t \*values, uint32\_t size, uint32\_t radix=10) const
- uint32\_t **getAttribute** (uint32\_t index, float32\_t \*values, uint32\_t size) const
- uint32\_t **getAttribute** (uint32\_t index, float64\_t \*values, uint32\_t size) const
- template<class Type >  
bool **getAttribute** (const char \*name, Type \*values, uint32\_t size) const
- template<class Type >  
uint32\_t **getAttribute** (uint32\_t index, Array< Type > &values) const
- template<class Type >  
uint32\_t **getAttribute** (const char \*name, Array< Type > &values) const

## 5.290.1 Detailed Description

[Xml](#)

## Index

- allocate
  - Tellusim::Allocator, [21](#)
- collapse
  - Tellusim::MeshReduce, [27](#)
- create
  - Tellusim::BitonicSort, [49](#)
  - Tellusim::CubeFilter, [146](#)
  - Tellusim::MeshGraph, [26](#)
  - Tellusim::MeshSolid, [28](#)
  - Tellusim::PrefixScan, [350](#)
  - Tellusim::RadixSort, [361](#)
  - Tellusim::SeparableFilter, [370](#)
  - Tellusim::SpatialTree, [385](#)
- create\_index\_buffer
  - Tellusim::MeshModel, [320](#)
- create\_meshlet\_buffer
  - Tellusim::MeshModel, [320](#)
- create\_vertex\_buffer
  - Tellusim::MeshModel, [320](#)
- createBasis
  - Tellusim::Mesh, [298](#)
  - Tellusim::MeshGeometry, [310](#)
- createBounds
  - Tellusim::Mesh, [298](#)
  - Tellusim::MeshGeometry, [309](#)
- createIslands
  - Tellusim::Mesh, [299](#)
  - Tellusim::MeshGeometry, [311](#)
- createNormals
  - Tellusim::Mesh, [299](#)
  - Tellusim::MeshGeometry, [310](#)
- createTangents
  - Tellusim::Mesh, [299](#)
  - Tellusim::MeshGeometry, [310](#)
- dispatch
  - Tellusim::BitonicSort, [50](#)
  - Tellusim::CubeFilter, [147](#)
  - Tellusim::DecoderJPEG, [176](#)
  - Tellusim::EncoderASTC, [196](#)
  - Tellusim::EncoderBC15, [198](#)
  - Tellusim::EncoderBC67, [199](#)
  - Tellusim::FourierTransform, [227](#)
  - Tellusim::PrefixScan, [351](#)
  - Tellusim::RadixSort, [361](#), [362](#)
  - Tellusim::SeparableFilter, [371](#)
  - Tellusim::SpatialGrid, [382](#)
  - Tellusim::SpatialTree, [385](#)
  - Tellusim::TensorGraph, [404](#)
- dispatchIndirect
  - Tellusim::BitonicSort, [50](#), [51](#)
  - Tellusim::PrefixScan, [352](#)
  - Tellusim::RadixSort, [362](#), [363](#)
  - Tellusim::SpatialGrid, [383](#)
  - Tellusim::SpatialTree, [386](#)
- dispatchYUV
  - Tellusim::DecoderJPEG, [176](#)
- exec
  - Tellusim::System, [35](#)
- float16x4\_t
  - Tellusim::float16x4\_t, [210](#)
- float16x8\_t
  - Tellusim::float16x8\_t, [212](#)
- float32x4\_t
  - Tellusim::float32x4\_t, [215](#)
- float32x8\_t
  - Tellusim::float32x8\_t, [217](#)
- getThreadID
  - Tellusim::System, [35](#)
- int16x8\_t
  - Tellusim::int16x8\_t, [271](#)
- int32x4\_t
  - Tellusim::int32x4\_t, [273](#)
- int32x8\_t
  - Tellusim::int32x8\_t, [275](#)
- load
  - Tellusim::DecoderJPEG, [175](#)
- loadLibrary
  - Tellusim::System, [35](#)
- loadTexture
  - Tellusim::DecoderJPEG, [176](#)
- normal
  - Tellusim::Polygon, [348](#)
- open
  - Tellusim::System, [36](#)
- optimizeAnimations
  - Tellusim::Mesh, [300](#)
- optimizeAttributes
  - Tellusim::MeshGeometry, [312](#)
- optimizeGeometries
  - Tellusim::Mesh, [300](#)
- optimizeIndices
  - Tellusim::MeshGeometry, [311](#)
- optimizeWinding
  - Tellusim::Mesh, [299](#)
- readRegion
  - Tellusim::Parser, [32](#)
- setEnvironment
  - Tellusim::System, [35](#)
- setInputSource
  - Tellusim::SeparableFilter, [370](#)
- setOutputSource
  - Tellusim::SeparableFilter, [370](#)

- setRange
  - Tellusim::ControlSlider, 137
- setString
  - Tellusim::Date, 174
- setValue
  - Tellusim::ControlSlider, 136
- skipDecimal
  - Tellusim::Parser, 31
- skipFloat
  - Tellusim::Parser, 31
- skipHexadecimal
  - Tellusim::Parser, 31
- skipLines
  - Tellusim::Parser, 33
- skipName
  - Tellusim::Parser, 31
- skipNumber
  - Tellusim::Parser, 32
- skipString
  - Tellusim::Parser, 32
- skipSymbol
  - Tellusim::Parser, 32
- skipToken
  - Tellusim::Parser, 30
- subdiv
  - Tellusim::MeshRefine, 27
- Tellusim::Allocator, 21
  - allocate, 21
- Tellusim::Android, 22
- Tellusim::App, 38
- Tellusim::Archive, 39
- Tellusim::ArchiveStream, 40
- Tellusim::Async, 40
- Tellusim::Async::Task, 399
- Tellusim::AsyncFunction< Func >, 42
- Tellusim::AsyncFunctionBase, 43
- Tellusim::AsyncFunctionRet< Type >, 43
- Tellusim::AsyncFunctionRet< void >, 44
- Tellusim::Atlas< Type >, 44
- Tellusim::Atlas< Type >::Node, 339
- Tellusim::AtomicArray< Type >, 45
- Tellusim::AtomicLock, 47
- Tellusim::AtomicPtr< Type >, 47
- Tellusim::Atomici32, 46
- Tellusim::Basis, 22
- Tellusim::BitonicSort, 48
  - create, 49
  - dispatch, 50
  - dispatchIndirect, 50, 51
- Tellusim::BitonicSort::DispatchParameters, 193
- Tellusim::Blob, 52
- Tellusim::BoundingBox< T, Vector3 >, 53
- Tellusim::BoundCircle< T, Vector2 >, 56
- Tellusim::BoundFrustum< T >, 58
- Tellusim::BoundRect< T, Vector2 >, 61
- Tellusim::BoundSphere< T, Vector3 >, 63
- Tellusim::BrepModel, 66
- Tellusim::BrepModel::FaceParameters, 201
- Tellusim::Buffer, 68
- Tellusim::BufferTable, 70
- Tellusim::CUBuffer, 147
- Tellusim::CUContext, 149
- Tellusim::CUSHader, 150
- Tellusim::CUTexture, 151
- Tellusim::Canvas, 71
- Tellusim::CanvasElement, 74
- Tellusim::CanvasEllipse, 77
- Tellusim::CanvasMesh, 79
- Tellusim::CanvasRect, 81
- Tellusim::CanvasShape, 82
- Tellusim::CanvasShapeVertex, 84
- Tellusim::CanvasStrip, 85
- Tellusim::CanvasStripVertex, 86
- Tellusim::CanvasText, 88
- Tellusim::CanvasTriangle, 89
- Tellusim::CanvasVertex, 91
- Tellusim::Color, 92
- Tellusim::Command, 94
- Tellusim::Command::DrawArraysIndirect, 194
- Tellusim::Command::DrawElementsIndirect, 195
- Tellusim::Command::DrawMeshIndirect, 195
- Tellusim::Compute, 98
- Tellusim::Compute::DispatchIndirect, 193
- Tellusim::Context, 101
- Tellusim::Control, 103
- Tellusim::ControlArea, 107
- Tellusim::ControlBase, 109
- Tellusim::ControlButton, 111
- Tellusim::ControlCheck, 113
- Tellusim::ControlCombo, 114
- Tellusim::ControlDialog, 117
- Tellusim::ControlEdit, 119
- Tellusim::ControlGrid, 121
- Tellusim::ControlGroup, 122
- Tellusim::ControlPanel, 125
- Tellusim::ControlRect, 126
- Tellusim::ControlRoot, 129
- Tellusim::ControlScroll, 132
- Tellusim::ControlSlider, 134
  - setRange, 137
  - setValue, 136
- Tellusim::ControlSplit, 137
- Tellusim::ControlText, 139
- Tellusim::ControlTree, 141
- Tellusim::ControlWindow, 144
- Tellusim::CubeFilter, 145
  - create, 146
  - dispatch, 147
- Tellusim::D3D11Buffer, 152
- Tellusim::D3D11Context, 153
- Tellusim::D3D11Device, 154
- Tellusim::D3D11Shader, 155
- Tellusim::D3D11Surface, 156
- Tellusim::D3D11Target, 157
- Tellusim::D3D11Texture, 158
- Tellusim::D3D12Buffer, 159

- Tellusim::D3D12Command, 160
- Tellusim::D3D12Compute, 161
- Tellusim::D3D12Context, 162
- Tellusim::D3D12Device, 163
- Tellusim::D3D12Kernel, 164
- Tellusim::D3D12Pipeline, 165
- Tellusim::D3D12Shader, 166
- Tellusim::D3D12Surface, 167
- Tellusim::D3D12Target, 169
- Tellusim::D3D12Texture, 170
- Tellusim::D3D12Tracing, 171
- Tellusim::D3D12Tracing::D3D12Instance, 164
- Tellusim::D3D12Traversal, 172
- Tellusim::Date, 173
  - setString, 174
- Tellusim::DecoderJPEG, 174
  - dispatch, 176
  - dispatchYUV, 176
  - load, 175
  - loadTexture, 176
- Tellusim::DefaultDestructor< Type >, 177
- Tellusim::Desktop, 177
- Tellusim::Device, 178
- Tellusim::Device::Features, 202
- Tellusim::DialogColor, 184
- Tellusim::DialogDirectory, 185
- Tellusim::DialogFileOpen, 185
- Tellusim::DialogFileSave, 186
- Tellusim::DialogMenu, 187
- Tellusim::DialogMessage, 189
- Tellusim::DialogProgress, 190
- Tellusim::Directory, 191
- Tellusim::Emscripten, 23
- Tellusim::EncoderASTC, 195
  - dispatch, 196
- Tellusim::EncoderBC15, 197
  - dispatch, 198
- Tellusim::EncoderBC67, 198
  - dispatch, 199
- Tellusim::Expression, 23
- Tellusim::FUBuffer, 228
- Tellusim::FUCommand, 229
- Tellusim::FUCompute, 230
- Tellusim::FUDevice, 231
- Tellusim::FUFence, 232
- Tellusim::FUKernel, 233
- Tellusim::FUPipeline, 234
- Tellusim::FUQuery, 235
- Tellusim::FUSampler, 236
- Tellusim::FUShader, 237
- Tellusim::FUTarget, 238
- Tellusim::FUTexture, 239
- Tellusim::FUTracing, 240
- Tellusim::FUTraversal, 241
- Tellusim::Face, 200
- Tellusim::Fence, 204
- Tellusim::File, 205
- Tellusim::FixedPool< Type, Index >, 206
- Tellusim::Font, 221
- Tellusim::FontBatch, 223
- Tellusim::FontBatch32, 224
- Tellusim::FontStyle, 225
- Tellusim::FourierTransform, 226
  - dispatch, 227
- Tellusim::GLBuffer, 242
- Tellusim::GLContext, 243
- Tellusim::GLESBuffer, 244
- Tellusim::GLESContext, 245
- Tellusim::GLESShader, 246
- Tellusim::GLESSurface, 247
- Tellusim::GLESTarget, 248
- Tellusim::GLESTexture, 249
- Tellusim::GLShader, 250
- Tellusim::GLSurface, 251
- Tellusim::GLTarget, 252
- Tellusim::GLTexture, 253
- Tellusim::GradientStyle, 254
- Tellusim::HIPBuffer, 256
- Tellusim::HIPContext, 257
- Tellusim::HIPShader, 258
- Tellusim::HIPTexture, 259
- Tellusim::HeapPool< Threshold >, 255
- Tellusim::Image, 259
- Tellusim::ImageColor, 264
- Tellusim::ImageSampler, 265
- Tellusim::ImageStream, 267
- Tellusim::Info, 268
- Tellusim::IsPtr< Type >, 275
- Tellusim::Json, 276
- Tellusim::Kernel, 279
- Tellusim::Layer, 282
- Tellusim::Log, 25
- Tellusim::LU, 25
- Tellusim::MTLBuffer, 325
- Tellusim::MTLCommand, 326
- Tellusim::MTLCompute, 327
- Tellusim::MTLContext, 328
- Tellusim::MTLDevice, 329
- Tellusim::MTLKernel, 330
- Tellusim::MTLPipeline, 331
- Tellusim::MTLSampler, 332
- Tellusim::MTLShader, 333
- Tellusim::MTLSurface, 334
- Tellusim::MTLTarget, 335
- Tellusim::MTLTexture, 336
- Tellusim::MTLTracing, 337
- Tellusim::MTLTracing::MTLInstance, 330
- Tellusim::Matrix3x2< T >, 282
- Tellusim::Matrix4x3< T >, 285
- Tellusim::Matrix4x4< T >, 289
- Tellusim::MatrixNxM< Type, N, M >, 293
- Tellusim::Mesh, 295
  - createBasis, 298
  - createBounds, 298
  - createIslands, 299
  - createNormals, 299



- createTangents, 299
- optimizeAnimations, 300
- optimizeGeometries, 300
- optimizeWinding, 299
- Tellusim::MeshAnimation, 300
- Tellusim::MeshAttachment, 302
- Tellusim::MeshAttribute, 304
- Tellusim::MeshGeometry, 306
  - createBasis, 310
  - createBounds, 309
  - createIslands, 311
  - createNormals, 310
  - createTangents, 310
  - optimizeAttributes, 312
  - optimizeIndices, 311
- Tellusim::MeshGraph, 26
  - create, 26
- Tellusim::MeshIndices, 312
- Tellusim::MeshJoint, 314
- Tellusim::MeshMaterial, 315
- Tellusim::MeshModel, 317
  - create\_index\_buffer, 320
  - create\_meshlet\_buffer, 320
  - create\_vertex\_buffer, 320
- Tellusim::MeshModel::Meshlet, 315
- Tellusim::MeshNode, 321
- Tellusim::MeshReduce, 26
  - collapse, 27
- Tellusim::MeshRefine, 27
  - subdiv, 27
- Tellusim::MeshSolid, 28
  - create, 28
- Tellusim::MeshStream, 322
- Tellusim::MeshTransform, 323
- Tellusim::MeshTransform::KeyData< Type >, 281
- Tellusim::Mipmap, 324
- Tellusim::Order, 29
- Tellusim::Origin, 340
- Tellusim::Parser, 29
  - readRegion, 32
  - skipDecimal, 31
  - skipFloat, 31
  - skipHexadecimal, 31
  - skipLines, 33
  - skipName, 31
  - skipNumber, 32
  - skipString, 32
  - skipSymbol, 32
  - skipToken, 30
- Tellusim::Pipeline, 340
- Tellusim::Polygon
  - normal, 348
  - triangulate, 348, 349
- Tellusim::Polygon< Capacity >, 347
- Tellusim::PrefixScan, 349
  - create, 350
  - dispatch, 351
  - dispatchIndirect, 352
- Tellusim::PrefixScan::DispatchParameters, 193
- Tellusim::QR, 33
- Tellusim::Quadrilateral, 33
- Tellusim::Quaternion< T >, 353
- Tellusim::Query, 356
- Tellusim::Query::Statistics, 388
- Tellusim::RadixCompare< float32\_t >, 358
- Tellusim::RadixCompare< int32\_t >, 358
- Tellusim::RadixCompare< Type >, 357
- Tellusim::RadixCompare< uint32\_t >, 358
- Tellusim::RadixMap< Key, Type, Size >, 358
- Tellusim::RadixMap< Key, Type, Size >::ConstIterator, 101
- Tellusim::RadixMap< Key, Type, Size >::Iterator, 276
- Tellusim::RadixSort, 359
  - create, 361
  - dispatch, 361, 362
  - dispatchIndirect, 362, 363
- Tellusim::RadixSort::DispatchParameters, 194
- Tellusim::Random< Integer, Float >, 363
- Tellusim::Rect, 364
- Tellusim::Region, 366
- Tellusim::SVD, 34
- Tellusim::Sampler, 366
- Tellusim::Scissor, 368
- Tellusim::SeparableFilter, 369
  - create, 370
  - dispatch, 371
  - setInputSource, 370
  - setOutputSource, 370
- Tellusim::Shader, 371
- Tellusim::ShaderCompiler, 375
- Tellusim::Size, 376
- Tellusim::Slice, 377
- Tellusim::Socket, 377
- Tellusim::SocketSSL, 379
- Tellusim::Source, 380
- Tellusim::Spatial, 33
- Tellusim::Spatial::Node< Type, Size >, 338
- Tellusim::SpatialGrid, 382
  - dispatch, 382
  - dispatchIndirect, 383
- Tellusim::SpatialGrid::DispatchParameters, 194
- Tellusim::SpatialTree, 383
  - create, 385
  - dispatch, 385
  - dispatchIndirect, 386
- Tellusim::SpatialTree::DispatchParameters, 194
- Tellusim::SpatialTree::Node, 340
- Tellusim::SpinLock, 387
- Tellusim::Stream, 388
- Tellusim::String, 391
- Tellusim::StrokeStyle, 394
- Tellusim::Surface, 396
- Tellusim::System, 35
  - exec, 35
  - getThreadID, 35
  - loadLibrary, 35



- open, 36
- setEnvironment, 35
- Tellusim::Target, 397
- Tellusim::Tensor, 400
- Tellusim::TensorGraph, 402
  - dispatch, 404
- Tellusim::Texture, 405
- Tellusim::TextureTable, 408
- Tellusim::Thread, 409
- Tellusim::ThreadClassFunction0< Class, Func >, 410
- Tellusim::ThreadClassFunction1< Class, Func, A0 >, 411
- Tellusim::ThreadClassFunction2< Class, Func, A0, A1 >, 412
- Tellusim::ThreadClassFunction3< Class, Func, A0, A1, A2 >, 413
- Tellusim::ThreadClassFunction4< Class, Func, A0, A1, A2, A3 >, 414
- Tellusim::ThreadFunction0< Func >, 415
- Tellusim::ThreadFunction1< Func, A0 >, 416
- Tellusim::ThreadFunction2< Func, A0, A1 >, 417
- Tellusim::ThreadFunction3< Func, A0, A1, A2 >, 418
- Tellusim::ThreadFunction4< Func, A0, A1, A2, A3 >, 419
- Tellusim::Time, 36
- Tellusim::Tracing, 420
- Tellusim::Tracing::BuildIndirect, 71
- Tellusim::Tracing::Instance, 269
- Tellusim::Traversal, 423
- Tellusim::Triangle, 36
- Tellusim::VKBuffer, 441
- Tellusim::VKCommand, 442
- Tellusim::VKCompute, 443
- Tellusim::VKContext, 444
- Tellusim::VKDevice, 445
- Tellusim::VKFence, 447
- Tellusim::VKShader, 448
- Tellusim::VKSurface, 449
- Tellusim::VKTarget, 450
- Tellusim::VKTexture, 451
- Tellusim::VKTracing, 452
- Tellusim::VKTracing::VKInstance, 447
- Tellusim::Vector2< T >, 431
- Tellusim::Vector3< T >, 433
- Tellusim::Vector4< T >, 436
- Tellusim::VectorN< Type, N >, 439
- Tellusim::Viewport, 440
- Tellusim::WinApp, 36
- Tellusim::Window, 453
- Tellusim::Windows, 37
- Tellusim::Xml, 459
- Tellusim::f32u32, 199
- Tellusim::f64u64, 200
- Tellusim::float16\_t, 207
- Tellusim::float16x4\_t, 209
  - float16x4\_t, 210
- Tellusim::float16x8\_t, 210
  - float16x8\_t, 212
- Tellusim::float21\_t, 212
- Tellusim::float24\_t, 213
- Tellusim::float32x4\_t, 214
  - float32x4\_t, 215
- Tellusim::float32x8\_t, 216
  - float32x8\_t, 217
- Tellusim::float64x2\_t, 217
- Tellusim::float64x4\_t, 218
- Tellusim::float64x8\_t, 220
- Tellusim::iOS, 24
- Tellusim::int16x8\_t, 269
  - int16x8\_t, 271
- Tellusim::int32x4\_t, 271
  - int32x4\_t, 273
- Tellusim::int32x8\_t, 273
  - int32x8\_t, 275
- Tellusim::uint16x8\_t, 426
  - uint16x8\_t, 427
- Tellusim::uint32x4\_t, 427
  - uint32x4\_t, 429
- Tellusim::uint32x8\_t, 429
  - uint32x8\_t, 431
- triangulate
  - Tellusim::Polygon, 348, 349
- uint16x8\_t
  - Tellusim::uint16x8\_t, 427
- uint32x4\_t
  - Tellusim::uint32x4\_t, 429
- uint32x8\_t
  - Tellusim::uint32x8\_t, 431