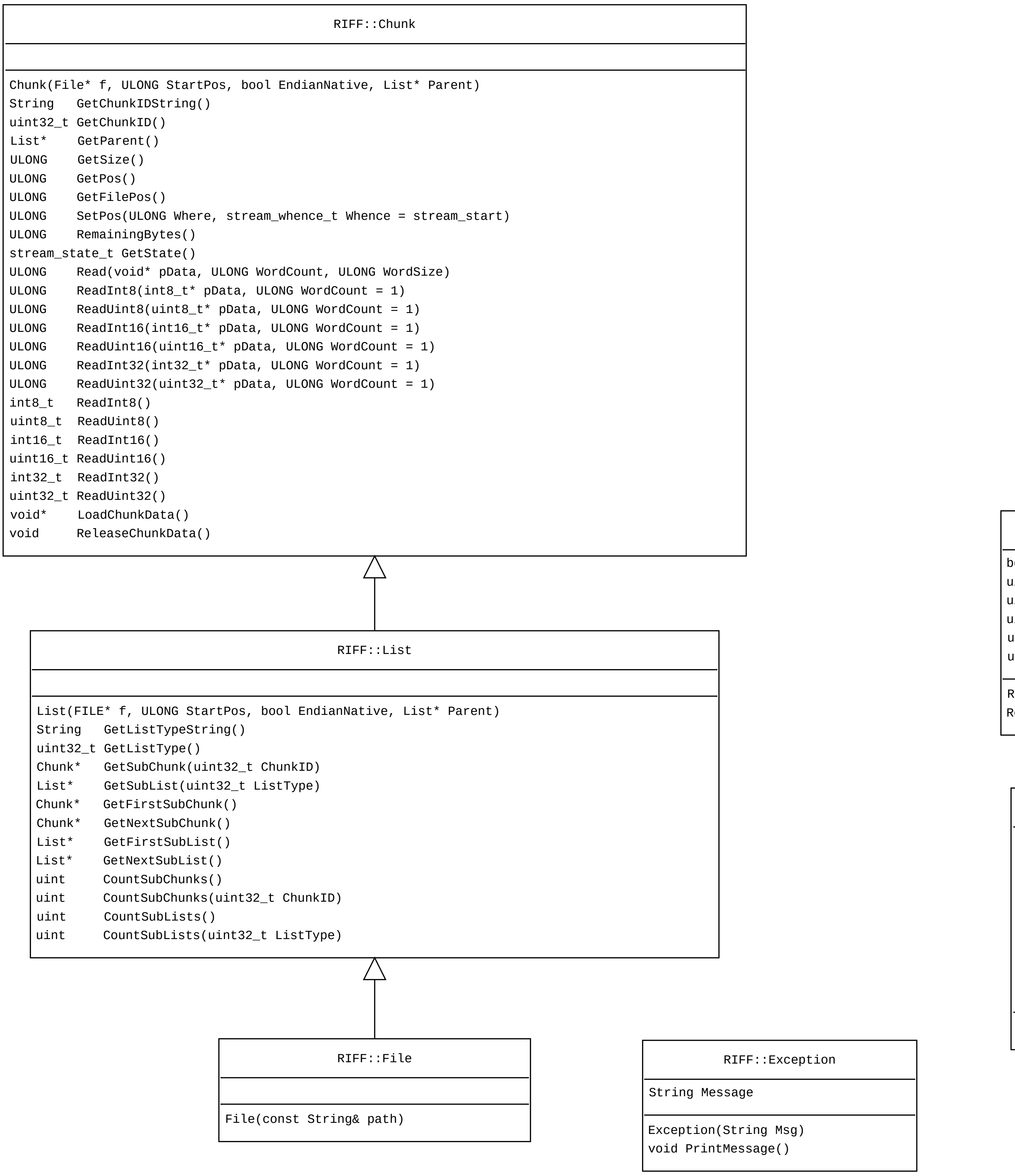
RIFF classes

If you just want to load a Gigasampler or DLS file, the only class here you should know about is the RIFF::File class. You have to provide an instantiation of that class to either the gig::File constructor or the DLS::File constructor.



```
RIFF::Chunk

Chunk(File* f, ULONG StartPos, bool EndianNative, List* Parent)
String   GetChunkIDString()
uint32_t GetChunkID()
List*    GetParent()
ULONG    GetSize()
ULONG    GetPos()
ULONG    GetFilePos()
ULONG    SetPos(ULONG Where, stream_whence_t Whence = stream_start)
ULONG    RemainingBytes()
stream_state_t GetState()
ULONG    Read(void* pData, ULONG WordCount, ULONG WordSize)
ULONG    ReadInt8(int8_t* pData, ULONG WordCount = 1)
ULONG    ReadUint8(uint8_t* pData, ULONG WordCount = 1)
ULONG    ReadInt16(int16_t* pData, ULONG WordCount = 1)
ULONG    ReadUint16(uint16_t* pData, ULONG WordCount = 1)
ULONG    ReadInt32(int32_t* pData, ULONG WordCount = 1)
ULONG    ReadUint32(uint32_t* pData, ULONG WordCount = 1)
int8_t   ReadInt8()
uint8_t  ReadUint8()
int16_t  ReadInt16()
uint16_t ReadUint16()
int32_t  ReadInt32()
uint32_t ReadUint32()
void*    LoadChunkData()
void     ReleaseChunkData()
```

```
RIFF::List

List(FILE* f, ULONG StartPos, bool EndianNative, List* Parent)
String   GetListTypeString()
uint32_t GetListType()
Chunk*   GetSubChunk(uint32_t ChunkID)
List*    GetSubList(uint32_t ListType)
Chunk*   GetFirstSubChunk()
Chunk*   GetNextSubChunk()
List*    GetFirstSubList()
List*    GetNextSubList()
uint     CountSubChunks()
uint     CountSubChunks(uint32_t ChunkID)
uint     CountSubLists()
uint     CountSubLists(uint32_t ListType)
```

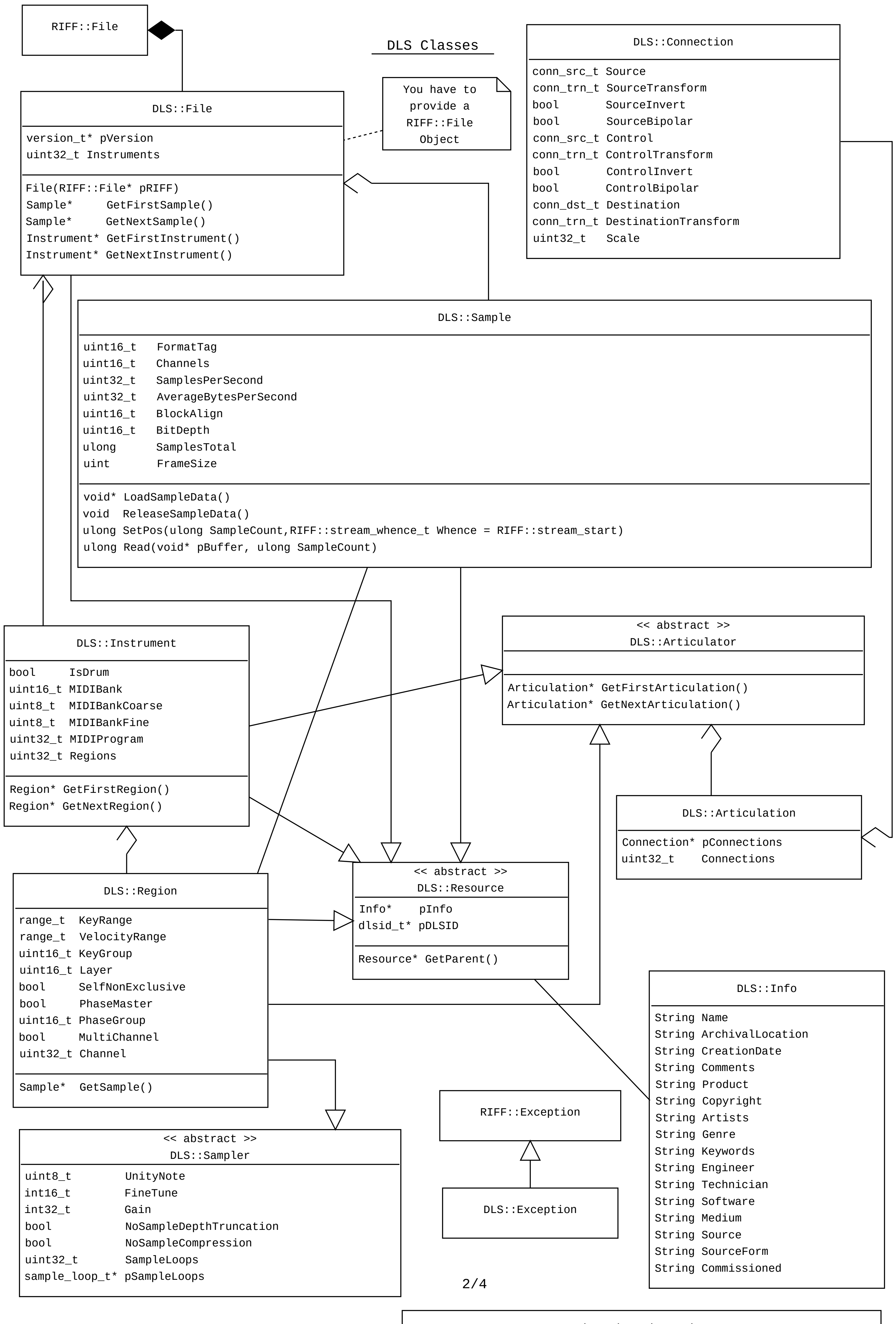
```
RIFF::File

File(const String& path)
```

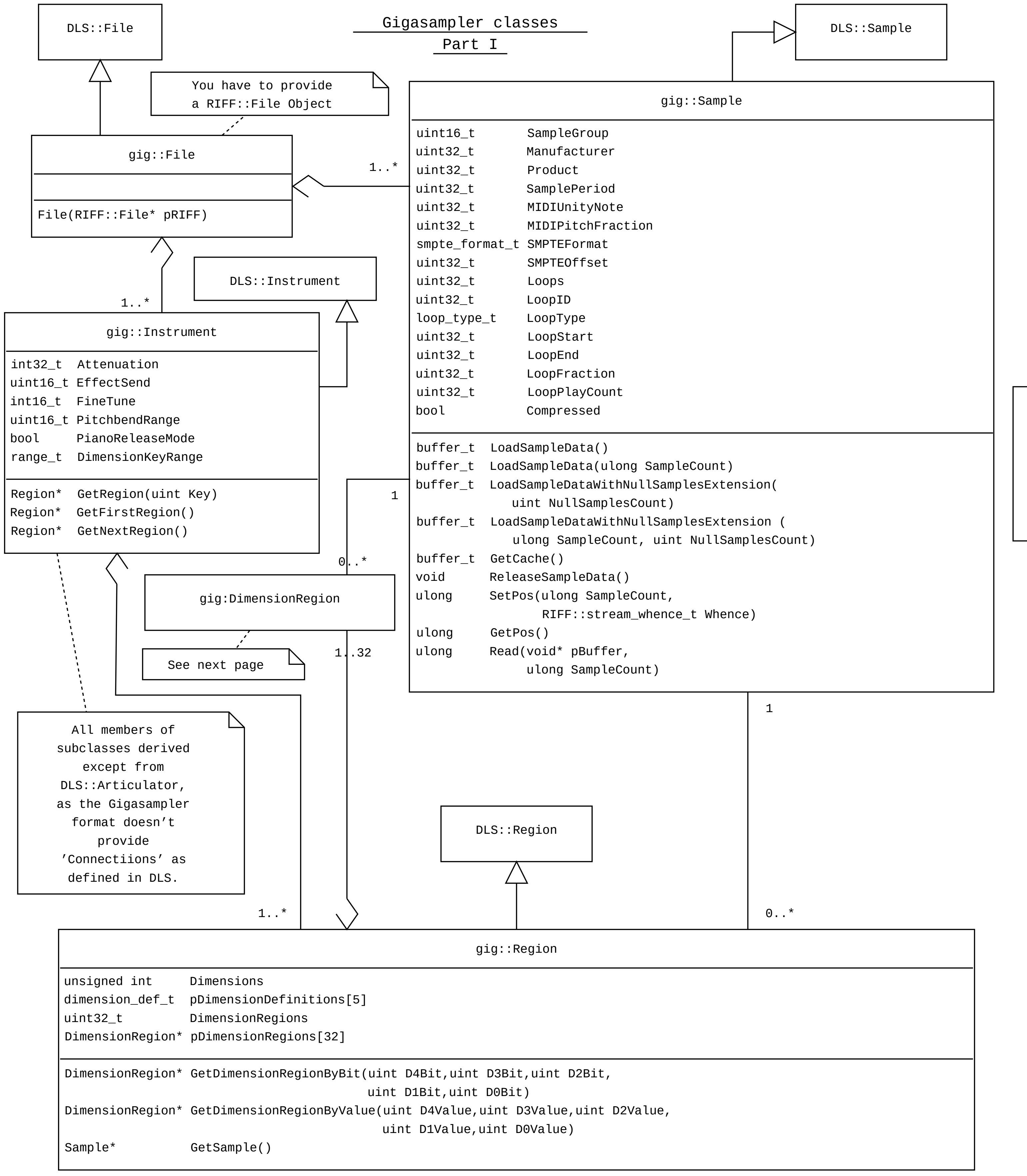
```
RIFF::Exception

String Message

Exception(String Msg)
void PrintMessage()
```


DLS Classes
RIFF::File
You have to provide a RIFF::File Object
DLS::File
version_t* pVersion
uint32_t Instruments
File(RIFF::File* pRIFF)
Sample* GetFirstSample()
Sample* GetNextSample()
Instrument* GetFirstInstrument()
Instrument* GetNextInstrument()
DLS::Connection
conn_src_t Source
conn_trn_t SourceTransform
bool SourceInvert
bool SourceBipolar
conn_src_t Control
conn_trn_t ControlTransform
bool ControlInvert
bool ControlBipolar
conn_dst_t Destination
conn_trn_t DestinationTransform
uint32_t Scale
DLS::Sample
uint16_t FormatTag
uint16_t Channels
uint32_t SamplesPerSecond
uint32_t AverageBytesPerSecond
uint16_t BlockAlign
uint16_t BitDepth
ulong SamplesTotal
uint FrameSize
void* LoadSampleData()
void ReleaseSampleData()
ulong SetPos(ulong SampleCount,RIFF::stream_whence_t Whence = RIFF::stream_start)
ulong Read(void* pBuffer, ulong SampleCount)
DLS::Instrument
bool IsDrum
uint16_t MIDIBank
uint8_t MIDIBankCoarse
uint8_t MIDIBankFine
uint32_t MIDIProgram
uint32_t Regions
Region* GetFirstRegion()
Region* GetNextRegion()
<< abstract >>
DLS::Articulator
Articulation* GetFirstArticulation()
Articulation* GetNextArticulation()
DLS::Articulation
Connection* pConnections
uint32_t Connections
<< abstract >>
DLS::Resource
Info* pInfo
dlsid_t* pDLSID
Resource* GetParent()
DLS::Region
range_t KeyRange
range_t VelocityRange
uint16_t KeyGroup
uint16_t Layer
bool SelfNonExclusive
bool PhaseMaster
uint16_t PhaseGroup
bool MultiChannel
uint32_t Channel
Sample* GetSample()
DLS::Info
String Name
String ArchivalLocation
String CreationDate
String Comments
String Product
String Copyright
String Artists
String Genre
String Keywords
String Engineer
String Technician
String Software
String Medium
String Source
String SourceForm
String Commissioned
RIFF::Exception
DLS::Exception
<< abstract >>
DLS::Sampler
uint8_t UnityNote
int16_t FineTune
int32_t Gain
bool NoSampleDepthTruncation
bool NoSampleCompression
uint32_t SampleLoops
sample_loop_t* pSampleLoops

Gigasampler classes
Part I



You have to provide
a RIFF::File Object

```
gig::File
-----------------------------------------
-----------------------------------------
File(RIFF::File* pRIFF)
```

```
gig::Sample
-----------------------------------------
uint16_t      SampleGroup
uint32_t      Manufacturer
uint32_t      Product
uint32_t      SamplePeriod
uint32_t      MIDIUnityNote
uint32_t      MIDIPitchFraction
smpte_format_t SMPTEFormat
uint32_t      SMPTEOffset
uint32_t      Loops
uint32_t      LoopID
loop_type_t   LoopType
uint32_t      LoopStart
uint32_t      LoopEnd
uint32_t      LoopFraction
uint32_t      LoopPlayCount
bool          Compressed
-----------------------------------------
buffer_t  LoadSampleData()
buffer_t  LoadSampleData(ulong SampleCount)
buffer_t  LoadSampleDataWithNullSamplesExtension(
             uint NullSamplesCount)
buffer_t  LoadSampleDataWithNullSamplesExtension (
             ulong SampleCount, uint NullSamplesCount)
buffer_t  GetCache()
void      ReleaseSampleData()
ulong     SetPos(ulong SampleCount,
                 RIFF::stream_whence_t Whence)
ulong     GetPos()
ulong     Read(void* pBuffer,
               ulong SampleCount)
```

```
gig::Instrument
-----------------------------------------
int32_t  Attenuation
uint16_t EffectSend
int16_t  FineTune
uint16_t PitchbendRange
bool     PianoReleaseMode
range_t  DimensionKeyRange
-----------------------------------------
Region*  GetRegion(uint Key)
Region*  GetFirstRegion()
Region*  GetNextRegion()
```

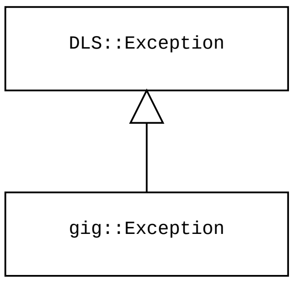
```
gig:DimensionRegion
```

See next page

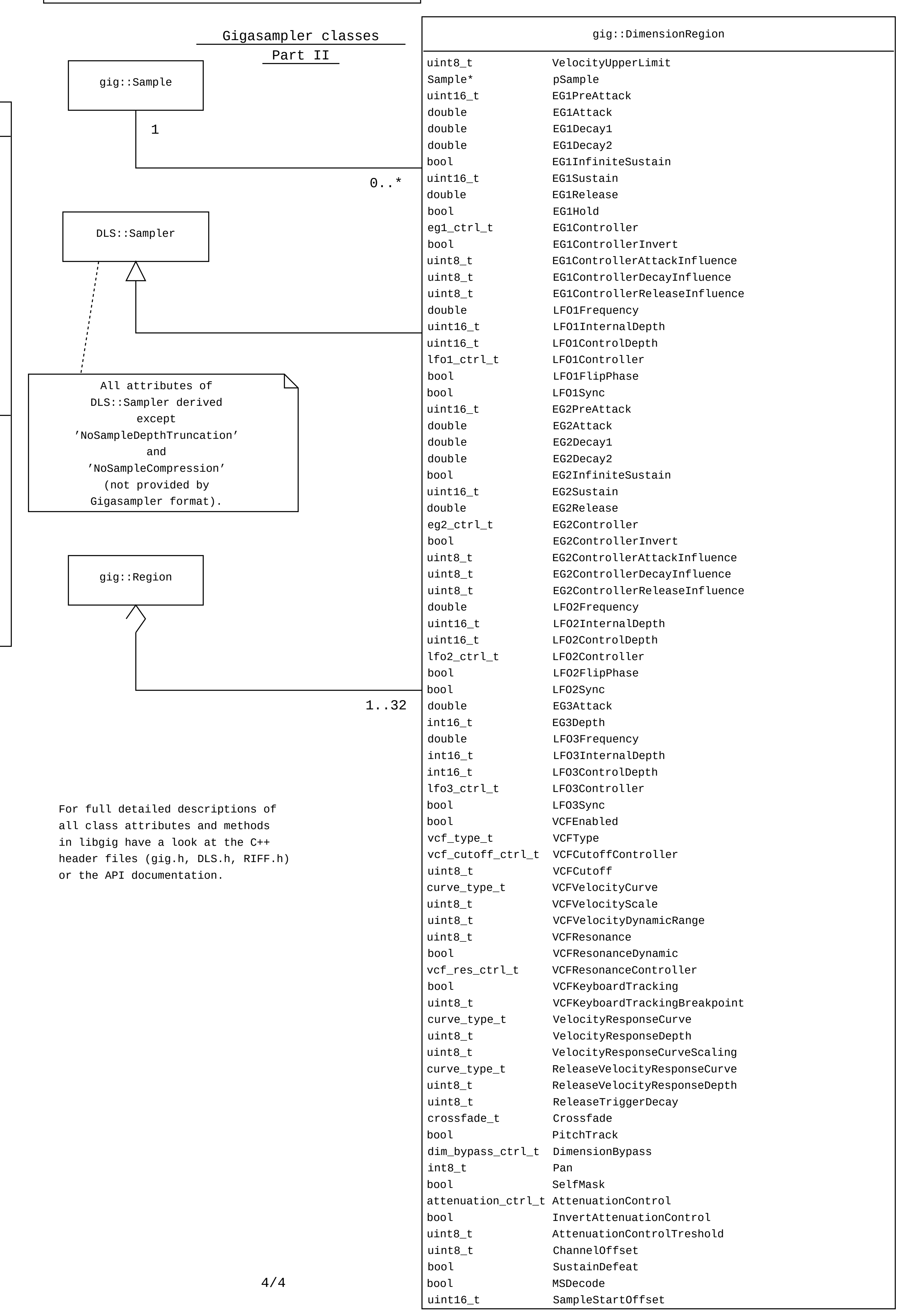
All members of subclasses derived except from DLS::Articulator, as the Gigasampler format doesn't provide 'Connectiions' as defined in DLS.

```
gig::Region
-----------------------------------------
unsigned int     Dimensions
dimension_def_t  pDimensionDefinitions[5]
uint32_t         DimensionRegions
DimensionRegion* pDimensionRegions[32]
-----------------------------------------
DimensionRegion* GetDimensionRegionByBit(uint D4Bit,uint D3Bit,uint D2Bit,
                                         uint D1Bit,uint D0Bit)
DimensionRegion* GetDimensionRegionByValue(uint D4Value,uint D3Value,uint D2Value,
                                           uint D1Value,uint D0Value)
Sample*          GetSample()
```

The Gigasampler classes are more or less just extensions of the DLS classes. So also have look at those derived DLS classes to get full overview of all available methods and class attributes the Gigasampler classes provide.



Gigasampler classes Part II



All attributes of DLS::Sampler derived except 'NoSampleDepthTruncation' and 'NoSampleCompression' (not provided by Gigasampler format).

gig::DimensionRegion

uint8_t	VelocityUpperLimit
Sample*	pSample
uint16_t	EG1PreAttack
double	EG1Attack
double	EG1Decay1
double	EG1Decay2
bool	EG1InfiniteSustain
uint16_t	EG1Sustain
double	EG1Release
bool	EG1Hold
eg1_ctrl_t	EG1Controller
bool	EG1ControllerInvert
uint8_t	EG1ControllerAttackInfluence
uint8_t	EG1ControllerDecayInfluence
uint8_t	EG1ControllerReleaseInfluence
double	LFO1Frequency
uint16_t	LFO1InternalDepth
uint16_t	LFO1ControlDepth
lfo1_ctrl_t	LFO1Controller
bool	LFO1FlipPhase
bool	LFO1Sync
uint16_t	EG2PreAttack
double	EG2Attack
double	EG2Decay1
double	EG2Decay2
bool	EG2InfiniteSustain
uint16_t	EG2Sustain
double	EG2Release
eg2_ctrl_t	EG2Controller
bool	EG2ControllerInvert
uint8_t	EG2ControllerAttackInfluence
uint8_t	EG2ControllerDecayInfluence
uint8_t	EG2ControllerReleaseInfluence
double	LFO2Frequency
uint16_t	LFO2InternalDepth
uint16_t	LFO2ControlDepth
lfo2_ctrl_t	LFO2Controller
bool	LFO2FlipPhase
bool	LFO2Sync
double	EG3Attack
int16_t	EG3Depth
double	LFO3Frequency
int16_t	LFO3InternalDepth
int16_t	LFO3ControlDepth
lfo3_ctrl_t	LFO3Controller
bool	LFO3Sync
bool	VCFEnabled
vcf_type_t	VCFType
vcf_cutoff_ctrl_t	VCFCutoffController
uint8_t	VCFCutoff
curve_type_t	VCFVelocityCurve
uint8_t	VCFVelocityScale
uint8_t	VCFVelocityDynamicRange
uint8_t	VCFResonance
bool	VCFResonanceDynamic
vcf_res_ctrl_t	VCFResonanceController
bool	VCFKeyboardTracking
uint8_t	VCFKeyboardTrackingBreakpoint
curve_type_t	VelocityResponseCurve
uint8_t	VelocityResponseDepth
uint8_t	VelocityResponseCurveScaling
curve_type_t	ReleaseVelocityResponseCurve
uint8_t	ReleaseVelocityResponseDepth
uint8_t	ReleaseTriggerDecay
crossfade_t	Crossfade
bool	PitchTrack
dim_bypass_ctrl_t	DimensionBypass
int8_t	Pan
bool	SelfMask
attenuation_ctrl_t	AttenuationControl
bool	InvertAttenuationControl
uint8_t	AttenuationControlTreshold
uint8_t	ChannelOffset
bool	SustainDefeat
bool	MSDecode
uint16_t	SampleStartOffset

For full detailed descriptions of all class attributes and methods in libgig have a look at the C++ header files (gig.h, DLS.h, RIFF.h) or the API documentation.