

**sosicon**

Version v 1.0 beta  
4/1/2016 11:19:00 AM



# Table of Contents

Module Index .....	2
Namespace Index .....	3
Hierarchical Index .....	4
Class Index .....	6
File Index .....	8
Module Documentation.....	10
Converters .....	10
Interfaces .....	10
SOSI Elements .....	11
sosicon.....	18
sosicon::byteOrder .....	22
sosicon::shape .....	25
sosicon::sosi .....	27
sosicon::sosi::chartables .....	29
sosicon::utils.....	30
Class Documentation .....	35
sosicon::CommandLine.....	35
sosicon::ConverterSosi2psql .....	42
sosicon::ConverterSosi2shp .....	50
sosicon::ConverterSosi2tsv .....	54
sosicon::ConverterSosi2xml.....	56
sosicon::ConverterSosiStat .....	58
sosicon::Coordinate .....	61
sosicon::CoordinateCollection .....	65
sosicon::sosi::CoordSys .....	71
sosicon::shape::DoubleField .....	73
sosicon::EventDispatcher< Event > .....	74
sosicon::Factory .....	76
sosicon::ConverterSosi2psql::Field.....	78
sosicon::IBinaryStreamable .....	80
sosicon::IConverter .....	82
sosicon::ICoordinate .....	84
sosicon::ILookupTable.....	87
imaxdiv_t .....	89
sosicon::shape::Int16Field.....	90
sosicon::shape::Int32Field.....	91
sosicon::shape::Int32TField .....	92
sosicon::shape::Int8Field.....	93
sosicon::IRectangle .....	94
sosicon::IShapeElement .....	97
sosicon::IShapeElementHeader.....	100
sosicon::IShapefile .....	101
sosicon::IShapefileDbfPart .....	103
sosicon::IShapefilePrjPart .....	105
sosicon::IShapefileShpPart .....	107
sosicon::IShapefileShxPart .....	109
sosicon::IShapeHeader .....	111
sosicon::ISosiElement .....	113
sosicon::ISosiHeadMember .....	116
sosicon::EventDispatcher< Event >::Listener .....	118
sosicon::LogEvent.....	119
sosicon::LogEventDispatcher.....	121

sosicon::Logger .....	122
sosicon::Parser .....	124
sosicon::sosi::ReferenceData .....	128
sosicon::shape::Shapefile .....	129
sosicon::shape::ShxIndex .....	143
sosicon::sosi::SosiCharsetSingleton .....	144
sosicon::sosi::SosiElement .....	148
sosicon::sosi::SosiElementSearch .....	154
sosicon::sosi::SosiJunctionPoint .....	157
sosicon::sosi::SosiNorthEast .....	159
sosicon::sosi::SosiOrigoNE .....	164
sosicon::sosi::SosiRefList .....	167
sosicon::sosi::SosiTranslationTable .....	169
sosicon::sosi::SosiUnit .....	172
File Documentation .....	174
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/byte_order.cpp .....	174
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/byte_order.h .....	175
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/command_line.cpp .....	176
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/command_line.h .....	177
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/common_types.h .....	178
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2psql.cpp .....	179
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2psql.h .....	180
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2shp.cpp .....	181
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2shp.h .....	182
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2tsv.cpp .....	183
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2tsv.h .....	184
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2xml.cpp .....	185
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2xml.h .....	186
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi_stat.cpp .....	187
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi_stat.h .....	188
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate.h .....	189
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate_collection.cpp .....	190
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate_collection.h .....	191
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/event_dispatcher.h .....	192
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/factory.cpp .....	193
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/factory.h .....	194
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_binary_streamable.h .....	195
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_converter.h .....	196
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_coordinate.h .....	197
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_lookup_table.h .....	198
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_rectangle.h .....	199
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shape_element.h .....	200
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shape_element_header.h .....	201
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shape_header.h .....	202
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile.h .....	203
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_dbf_part.h .....	204
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_prj_part.h .....	205
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_shp_part.h .....	206
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_shx_part.h .....	207
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_sosi_element.h .....	208
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_sosi_head_member.h .....	209
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/inttypes.h .....	210
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/log_event.h .....	230
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/logger.cpp .....	231
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/logger.h .....	232
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/main.cpp .....	233

/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/main.h.....	234
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/parser.cpp .....	235
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/parser.h.....	236
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/parser_ragel.cpp .....	237
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/parser.rl .....	238
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/sosi_north_east.rl .....	239
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/sosi_north_east_height.rl .....	240
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/sosi_origo_ne.rl .....	241
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/sosi_ref.rl .....	242
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile.cpp .....	243
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile.h .....	244
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile_types.h.....	245
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_charset_singleton.cpp .....	246
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_charset_singleton.h .....	247
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_element.cpp.....	248
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_element.h .....	249
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_element_search.cpp.....	250
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_element_search.h .....	251
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_junction_point.h .....	252
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_north_east.cpp.....	253
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_north_east.h .....	254
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_origo_ne.cpp .....	255
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_origo_ne.h.....	256
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_ref_list.cpp .....	257
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_ref_list.h.....	258
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_translation_table.cpp .....	259
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_translation_table.h.....	260
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_types.h.....	261
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_unit.cpp .....	263
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_unit.h.....	264
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi_north_east_height_ragel.cpp .....	265
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi_north_east_ragel.cpp .....	266
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi_origo_ne_ragel.cpp.....	267
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi_ref_ragel.cpp .....	268
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/utils.cpp.....	269
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/utils.h.....	270
Index .....	272



# Module Index

## Modules

Here is a list of all modules:

Converters .....	10
Interfaces .....	10
SOSI Elements .....	11

# Namespace Index

## Namespace List

Here is a list of all namespaces with brief descriptions:

<b>sosicon (Application root )</b>	18
<b>sosicon::byteOrder (Big/low-endian conversions )</b>	22
<b>sosicon::shape (ESRI Shape )</b>	25
<b>sosicon::sosi (SOSI )</b>	27
<b>sosicon::sosi::chartables</b>	29
<b>sosicon::utils (String manipulation routines )</b>	30



# Hierarchical Index

## Class Hierarchy

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

sosicon::CommandLine .....	35
sosicon::CoordinateCollection .....	65
sosicon::sosi::CoordSys .....	71
sosicon::shape::DoubleField .....	73
sosicon::EventDispatcher< Event > .....	74
sosicon::EventDispatcher< LogEvent > .....	74
sosicon::LogEventDispatcher .....	121
sosicon::Factory .....	76
sosicon::ConverterSosi2psql::Field .....	78
sosicon::IBinaryStreamable .....	80
sosicon::IShapeElement .....	97
sosicon::IShapeElementHeader .....	100
sosicon::IShapefileDbfPart .....	103
sosicon::IShapefile .....	101
sosicon::shape::Shapefile .....	129
sosicon::IShapefilePrjPart .....	105
sosicon::IShapefile .....	101
sosicon::IShapefileShpPart .....	107
sosicon::IShapefile .....	101
sosicon::IShapefileShxPart .....	109
sosicon::IShapefile .....	101
sosicon::IShapeHeader .....	111
sosicon::IConverter .....	82
sosicon::ConverterSosi2psql .....	42
sosicon::ConverterSosi2shp .....	50
sosicon::ConverterSosi2tsv .....	54
sosicon::ConverterSosi2xml .....	56
sosicon::ConverterSosiStat .....	58
sosicon::ICoordinate .....	84
sosicon::Coordinate .....	61
sosicon::ILookupTable .....	87
imaxdiv_t .....	89

sosicon::shape::Int16Field .....	90
sosicon::shape::Int32Field .....	91
sosicon::shape::Int32TField .....	92
sosicon::shape::Int8Field .....	93
sosicon::IRectangle .....	94
sosicon::ISosiElement .....	113
sosicon::sosi::SosiElement .....	148
 sosicon::ISosiHeadMember .....	116
sosicon::sosi::SosiCharsetSingleton .....	144
sosicon::sosi::SosiOrigoNE .....	164
sosicon::sosi::SosiUnit .....	172
 sosicon::EventDispatcher< Event >::Listener .....	118
sosicon::LogEvent .....	119
sosicon::Logger .....	122
sosicon::Parser .....	124
sosicon::sosi::ReferenceData .....	128
sosicon::shape::ShxIndex .....	143
sosicon::sosi::SosiElementSearch .....	154
sosicon::sosi::SosiJunctionPoint .....	157
sosicon::sosi::SosiNorthEast .....	159
sosicon::sosi::SosiRefList .....	167
sosicon::sosi::SosiTranslationTable .....	169

# Class Index

## Class List

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

<b>sosicon::CommandLine (Command-line parser )</b>	35
<b>sosicon::ConverterSosi2psql (SOSI to PostgreSQL/PostGIS converter )</b>	42
<b>sosicon::ConverterSosi2shp (SOSI to ESRI Shape converter )</b>	50
<b>sosicon::ConverterSosi2tsv (SOSI to TSV converter )</b>	54
<b>sosicon::ConverterSosi2xml (SOSI to ESRI Shape converter )</b>	56
<b>sosicon::ConverterSosiStat (SOSI to ESRI Shape converter )</b>	58
<b>sosicon::Coordinate (Coordinate container )</b>	61
<b>sosicon::CoordinateCollection (Coordinate container )</b>	65
<b>sosicon::sosi::CoordSys (SOSI coordinate system )</b>	71
<b>sosicon::shape::DoubleField (32 bit double / byte field )</b>	73
<b>sosicon::EventDispatcher&lt; Event &gt; (Event dispatcher template class )</b>	74
<b>sosicon::Factory (Factory class )</b>	76
<b>sosicon::ConverterSosi2psql::Field</b>	78
<b>sosicon::IBinaryStreamable (Interface: Binary streamable object )</b>	80
<b>sosicon::IConverter (Interface: Converter )</b>	82
<b>sosicon::ICoordinate (Interface: Coordinate )</b>	84
<b>sosicon::ILookupTable (Interface: Lookup table )</b>	87
<b>imaxdiv_t</b>	89
<b>sosicon::shape::Int16Field (16 bit integer / byte field )</b>	90
<b>sosicon::shape::Int32Field (32 bit integer / byte field )</b>	91
<b>sosicon::shape::Int32TField (32 bit integer / byte / geom::ShapeType field )</b>	92
<b>sosicon::shape::Int8Field (8 bit integer / byte field )</b>	93
<b>sosicon::IRectangle (Interface: Rectangle )</b>	94
<b>sosicon::IShapeElement (Interface: Shape element )</b>	97
<b>sosicon::IShapeElementHeader (Interface: Shape element header )</b>	100
<b>sosicon::IShapefile (Interface: Shapefile )</b>	101
<b>sosicon::IShapefileDbfPart (Interface: ShapefileDbfPart )</b>	103
<b>sosicon::IShapefilePrjPart (Interface: ShapefilePrjPart )</b>	105
<b>sosicon::IShapefileShpPart (Interface: ShapefileShpPart )</b>	107
<b>sosicon::IShapefileShxPart (Interface: ShapefileShxPart )</b>	109
<b>sosicon::IShapeHeader (Interface: Shape element )</b>	111
<b>sosicon::ISosiElement (Interface: SOSI element )</b>	113
<b>sosicon::ISosiHeadMember (Interface: SOSI header element )</b>	116
<b>sosicon::EventDispatcher&lt; Event &gt;::Listener</b>	118
<b>sosicon::LogEvent (Log event )</b>	119
<b>sosicon::LogEventDispatcher</b>	121
<b>sosicon::Logger (SOSI logger )</b>	122
<b>sosicon::Parser (SOSI file parser )</b>	124
<b>sosicon::sosi::ReferenceData (SOSI reference number )</b>	128
<b>sosicon::shape::Shapefile (Shapefile implementation )</b>	129

<b>sosicon::shape::ShxIndex</b> .....	143
<b>sosicon::sosi::SosiCharsetSingleton (SOSI Character set )</b> .....	144
<b>sosicon::sosi::SosiElement (Basic SOSI element )</b> .....	148
<b>sosicon::sosi::SosiElementSearch</b> .....	154
<b>sosicon::sosi::SosiJunctionPoint (SOSI Junction point )</b> .....	157
<b>sosicon::sosi::SosiNorthEast (SOSI North-east element )</b> .....	159
<b>sosicon::sosi::SosiOrigoNE (SOSI Junction point )</b> .....	164
<b>sosicon::sosi::SosiRefList (SOSI REF list )</b> .....	167
<b>sosicon::sosi::SosiTranslationTable</b> .....	169
<b>sosicon::sosi::SosiUnit (SOSI Unit )</b> .....	172

# File Index

## File List

Here is a list of all files with brief descriptions:

/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/byte_order.cpp .....	174
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/byte_order.h .....	175
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/command_line.cpp .....	176
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/command_line.h .....	177
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/common_types.h .....	178
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2psql.cpp .....	179
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2psql.h .....	180
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2shp.cpp .....	181
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2shp.h .....	182
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2tsv.cpp .....	183
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2tsv.h .....	184
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2xml.cpp .....	185
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi2xml.h .....	186
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi_stat.cpp .....	187
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter_sosi_stat.h .....	188
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate.h .....	189
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate_collection.cpp .....	190
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate_collection.h .....	191
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/event_dispatcher.h .....	192
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/factory.cpp .....	193
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/factory.h .....	194
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/inttypes.h .....	210
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/log_event.h .....	230
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/logger.cpp .....	231
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/logger.h .....	232
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/main.cpp .....	233
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/main.h .....	234
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/parser.cpp .....	235
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/parser.h .....	236
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/parser_ragel.cpp .....	237
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi_north_east_height_ragel.cpp ..	265
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi_north_east_ragel.cpp .....	266
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi_origo_ne_ragel.cpp .....	267
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi_ref_ragel.cpp .....	268
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/utils.cpp .....	269
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/utils.h .....	270
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_binary_streamable.h ....	195
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_converter.h .....	196
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_coordinate.h .....	197
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_lookup_table.h .....	198

/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_rectangle.h .....	199
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shape_element.h .....	200
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shape_element_header.h .....	201
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shape_header.h .....	202
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile.h .....	203
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_dbf_part.h ....	204
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_prj_part.h ....	205
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_shp_part.h ....	206
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_shx_part.h ....	207
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_sosi_element.h .....	208
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_sosi_head_member.h ....	209
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/parser.rl .....	238
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/sosi_north_east.rl .....	239
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/sosi_north_east_height.rl .....	240
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/sosi_origo_ne.rl .....	241
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/sosi_ref.rl .....	242
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile.cpp .....	243
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile.h .....	244
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile_types.h .....	245
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_charset_singleton.cpp .....	246
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_charset_singleton.h .....	247
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_element.cpp .....	248
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_element.h .....	249
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_element_search.cpp .....	250
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_element_search.h .....	251
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_junction_point.h .....	252
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_north_east.cpp .....	253
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_north_east.h .....	254
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_origo_ne.cpp .....	255
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_origo_ne.h .....	256
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_ref_list.cpp .....	257
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_ref_list.h .....	258
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_translation_table.cpp .....	259
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_translation_table.h .....	260
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_types.h .....	261
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_unit.cpp .....	263
/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_unit.h .....	264

# Module Documentation

## Converters

### Classes

- class **sosicon::ConverterSosi2psql**
  - *SOSI to PostgreSQL/PostGIS converter.* class **sosicon::ConverterSosi2shp**
  - *SOSI to ESRI Shape converter.* class **sosicon::ConverterSosi2tsv**
  - *SOSI to TSV converter.* class **sosicon::ConverterSosi2xml**
  - *SOSI to ESRI Shape converter.* class **sosicon::ConverterSosiStat**
- SOSI to ESRI Shape converter.*
- 

### Detailed Description

Collection of **sosicon::IConverter** implementations for handling and generating file output. Different command-line arguments will employ different converters. The factory class **sosicon::Factory** is responsible for creating correct **IConverter** instance. The following arguments are currently interpreted:

- -2shp: **sosicon::ConverterSosi2shp** Shapefile conversion
- -2tsv: **sosicon::ConverterSosi2tsv** TSV file conversion
- -2xml: **sosicon::ConverterSosi2xml** Shape file conversion
- -stat: **sosicon::ConverterSosiStat** SOSI statistics (printout)

Collection of **sosicon::IConverter** implementations for handling and generating file output. Different command-line arguments will employ different converters. The factory class **sosicon::Factory** is responsible for creating correct **IConverter** instance. The following arguments are currently interpreted:

- -2shp: **sosicon::ConverterSosi2shp** Shapefile conversion
- -2tsv: **sosicon::ConverterSosi2tsv** TSV file conversion
- -2xml: **sosicon::ConverterSosi2Xml** XML conversion
- -2psql: **sosicon::ConverterSosi2Psql** PostgreSQL dump file conversion
- -stat: **sosicon::ConverterSosiStat** SOSI statistics (printout)

## Interfaces

### Classes

- class **sosicon::IBinaryStreamable**
- *Interface: Binary streamable object.* class **sosicon::IConverter**
- *Interface: Converter.* class **sosicon::ICoordinate**
- *Interface: **Coordinate**.* class **sosicon::ILookupTable**
- *Interface: Lookup table.* class **sosicon::IRectangle**
- *Interface: Rectangle.* class **sosicon::IShapeElement**
- *Interface: Shape element.* class **sosicon::IShapeElementHeader**
- *Interface: Shape element header.* class **sosicon::IShapeHeader**
- *Interface: Shape element.* class **sosicon::IShapefile**
- *Interface: Shapefile.* class **sosicon::IShapefileDbfPart**
- *Interface: ShapefileDbfPart.* class **sosicon::IShapefilePrjPart**
- *Interface: ShapefilePrjPart.* class **sosicon::IShapefileShpPart**
- *Interface: ShapefileShpPart.* class **sosicon::IShapefileShxPart**
- *Interface: ShapefileShxPart.* class **sosicon::ISosiElement**
- *Interface: SOSI element.* class **sosicon::ISosiHeadMember**

## Interface: SOSI header element. Functions

- `std::ostream & sosicon::operator<< (std::ostream &os, IBinaryStreamable &binaryStreamable)`  
*Stream output operator.*

---

## Detailed Description

This is a listing of generic interfaces used within sosicon.

---

## Function Documentation

**`std::ostream& sosicon::operator<< (std::ostream & os, IBinaryStreamable & binaryStreamable)[inline]`**

Stream output operator.

Placed outside any class definitions. Invokes WriteBinary() on the source **IBinaryStreamable** object.

### Parameters:

<i>os</i>	target stream.
<i>binaryStreamable</i>	target object.

### Returns:

The stream object is returned to allow for chaining.

Definition at line 58 of file `i_binary_streamable.h`.

## SOSI Elements

### Classes

- class **`sosicon::sosi::SosiCharsetSingleton`**
- *SOSI Character set.* class **`sosicon::sosi::SosiElement`**
- *Basic SOSI element.* class **`sosicon::sosi::SosiElementSearch`**
- class **`sosicon::sosi::SosiJunctionPoint`**
- *SOSI Junction point.* class **`sosicon::sosi::SosiNorthEast`**
- *SOSI North-east element.* class **`sosicon::sosi::SosiOrigoNE`**
- *SOSI Junction point.* class **`sosicon::sosi::SosiRefList`**
- *SOSI REF list.* class **`sosicon::sosi::SosiTranslationTable`**
- struct **`sosicon::sosi::ReferenceData`**
- *SOSI reference number.* class **`sosicon::sosi::CoordSys`**
- *SOSI coordinate system.* class **`sosicon::sosi::SosiUnit`**

### SOSI Unit. Typedefs

- typedef `std::map< std::string, ISosiElement * >` **`sosicon::sosi::SosiElementMap`**  
*Element index type.*
- typedef `std::vector< ISosiElement * >` **`sosicon::sosi::SosiChildrenList`**
- typedef `SosiChildrenList::iterator` **`sosicon::sosi::SosiChildrenIterator`**
- typedef `std::vector< ReferenceData * >` **`sosicon::sosi::GeometryRef`**



*List of SOSI references.*

- typedef std::vector< GeometryRef \* > **sosicon::sosi::GeometryCollection**  
*Collection of SOSI reference lists.*

## Enumerations

- enum **sosicon::sosi::ElementType** { **sosicon::sosi::sosi\_element\_unknown**,  
**sosicon::sosi::sosi\_element\_address\_identifier**, **sosicon::sosi::sosi\_element\_airport\_roads**,  
**sosicon::sosi::sosi\_element\_airport\_type**, **sosicon::sosi::sosi\_element\_area**,  
**sosicon::sosi::sosi\_element\_charset**, **sosicon::sosi::sosi\_element\_coordsys**,  
**sosicon::sosi::sosi\_element\_curve**, **sosicon::sosi::sosi\_element\_data\_collection\_date**,  
**sosicon::sosi::sosi\_element\_eof**, **sosicon::sosi::sosi\_element\_head**, **sosicon::sosi::sosi\_element\_height**,  
**sosicon::sosi::sosi\_element\_iata\_code**, **sosicon::sosi::sosi\_element\_icao\_code**,  
**sosicon::sosi::sosi\_element\_kp**, **sosicon::sosi::sosi\_element\_level**, **sosicon::sosi::sosi\_element\_max\_ne**,  
**sosicon::sosi::sosi\_element\_min\_ne**, **sosicon::sosi::sosi\_element\_municipality**,  
**sosicon::sosi::sosi\_element\_name**, **sosicon::sosi::sosi\_element\_ne**, **sosicon::sosi::sosi\_element\_neh**,  
**sosicon::sosi::sosi\_element\_objtype**, **sosicon::sosi::sosi\_element\_origo\_ne**,  
**sosicon::sosi::sosi\_element\_owner**, **sosicon::sosi::sosi\_element\_point**, **sosicon::sosi::sosi\_element\_quality**,  
**sosicon::sosi::sosi\_element\_ref**, **sosicon::sosi::sosi\_element\_surface**, **sosicon::sosi::sosi\_element\_text**,  
**sosicon::sosi::sosi\_element\_traffic\_type**, **sosicon::sosi::sosi\_element\_transpar**,  
**sosicon::sosi::sosi\_element\_unit**, **sosicon::sosi::sosi\_element\_updatedate**,  
**sosicon::sosi::sosi\_element\_water\_width**, **sosicon::sosi::sosi\_element\_vendor**,  
**sosicon::sosi::sosi\_element\_version** } *List of SOSI element types.*
- enum **sosicon::sosi::ObjType** { **sosicon::sosi::sosi\_objtype\_unknown**, **sosicon::sosi::sosi\_objtype\_airport**,  
**sosicon::sosi::sosi\_objtype\_airport\_type**, **sosicon::sosi::sosi\_objtype\_baseline**,  
**sosicon::sosi::sosi\_objtype\_carriageway**, **sosicon::sosi::sosi\_objtype\_cadastral\_address**,  
**sosicon::sosi::sosi\_objtype\_coastline**, **sosicon::sosi::sosi\_objtype\_constituency\_boundary**,  
**sosicon::sosi::sosi\_objtype\_county\_boundary**, **sosicon::sosi::sosi\_objtype\_data\_delineation**,  
**sosicon::sosi::sosi\_objtype\_edge\_view**, **sosicon::sosi::sosi\_objtype\_fictitious\_dividing\_line**,  
**sosicon::sosi::sosi\_objtype\_forest**, **sosicon::sosi::sosi\_objtype\_developed\_area**,  
**sosicon::sosi::sosi\_objtype\_golf\_course**, **sosicon::sosi::sosi\_objtype\_industrial\_area**,  
**sosicon::sosi::sosi\_objtype\_lake**, **sosicon::sosi::sosi\_objtype\_lane**, **sosicon::sosi::sosi\_objtype\_lake\_edge**,  
**sosicon::sosi::sosi\_objtype\_lake\_river\_barrier**, **sosicon::sosi::sosi\_objtype\_land\_use\_boundary**,  
**sosicon::sosi::sosi\_objtype\_level\_crossing**, **sosicon::sosi::sosi\_objtype\_municipal\_divide**,  
**sosicon::sosi::sosi\_objtype\_municipality**, **sosicon::sosi::sosi\_objtype\_municipality\_boundary**,  
**sosicon::sosi::sosi\_objtype\_marsh**, **sosicon::sosi::sosi\_objtype\_national\_border**,  
**sosicon::sosi::sosi\_objtype\_pedestrian\_bicycle\_road\_centre\_line**,  
**sosicon::sosi::sosi\_objtype\_sea\_river\_delineation**, **sosicon::sosi::sosi\_objtype\_snow\_field**,  
**sosicon::sosi::sosi\_objtype\_open\_land**, **sosicon::sosi::sosi\_objtype\_river\_brook**,  
**sosicon::sosi::sosi\_objtype\_river\_brook\_edge**, **sosicon::sosi::sosi\_objtype\_road\_block**,  
**sosicon::sosi::sosi\_objtype\_road\_centre\_line**, **sosicon::sosi::sosi\_objtype\_road\_under\_railway**,  
**sosicon::sosi::sosi\_objtype\_sea\_surface**, **sosicon::sosi::sosi\_objtype\_sidewalk**,  
**sosicon::sosi::sosi\_objtype\_spelling**, **sosicon::sosi::sosi\_objtype\_stone\_quarry**,  
**sosicon::sosi::sosi\_objtype\_street\_address**, **sosicon::sosi::sosi\_objtype\_territorial\_boundary**,  
**sosicon::sosi::sosi\_objtype\_turn\_connecting\_segment** } *List of SOSI OBJTYPEs.*
- enum **sosicon::sosi::Charset** { **sosicon::sosi::sosi\_charset\_undetermined**, **sosicon::sosi::sosi\_charset\_ansi**,  
**sosicon::sosi::sosi\_charset\_decn7**, **sosicon::sosi::sosi\_charset\_dosn8**,  
**sosicon::sosi::sosi\_charset\_iso8859\_1**, **sosicon::sosi::sosi\_charset\_iso8859\_10**,  
**sosicon::sosi::sosi\_charset\_nd7**, **sosicon::sosi::sosi\_charset\_utf8** } *SOSI character encodings.*
- enum **sosicon::sosi::JunctionPoint** { **sosicon::sosi::sosi\_junction\_node**,  
**sosicon::sosi::sosi\_junction\_connection**, **sosicon::sosi::sosi\_junction\_open\_end** } *Default SOSI junction point layer types.*

## Functions

- CoordSys **sosicon::sosi::sysCodeToCoordSys** (int sysCode)

*Convert SOSI SYSKODE value to coordinate system data.*

- ElementType **sosicon::sosi::sosiNameToType** (std::string sosiElementName)  
*Convert SOSI element names to ElementType enum value.*
- ObjType **sosicon::sosi::sosiObjNameToType** (std::string sosiObjTypeName)  
*Convert SOSI objtype names to ObjType enum value.*

---

## Detailed Description

Implemented representation of SOSI file elements.

---

## Typedef Documentation

**typedef std::vector<GeometryRef\*> sosicon::sosi::GeometryCollection**

Collection of SOSI reference lists.

Definition at line 185 of file sosi\_types.h.

**typedef std::vector<ReferenceData\*> sosicon::sosi::GeometryRef**

List of SOSI references.

Definition at line 182 of file sosi\_types.h.

**typedef SosiChildrenList::iterator sosicon::sosi::SosiChildrenIterator**

Definition at line 44 of file sosi\_element\_search.h.

**typedef std::vector<ISosiElement\*> sosicon::sosi::SosiChildrenList**

Definition at line 42 of file sosi\_element\_search.h.

**typedef std::map<std::string,ISosiElement\*> sosicon::sosi::SosiElementMap**

Element index type.

Definition at line 40 of file sosi\_element\_search.h.

---

## Enumeration Type Documentation

**enum sosicon::sosi::Charset**

SOSI character encodings.

**Enumerator**

*sosi\_charset\_undetermined* Charset element not yet encountered.

*sosi\_charset\_ansi* ANSI; equals ISO8859-1.  
*sosi\_charset\_decn7* Dec Norwegian 7-bit.  
*sosi\_charset\_dosn8* MS-Dos Norwegian 8-bit.  
*sosi\_charset\_iso8859\_1* ISO 8859-1.  
*sosi\_charset\_iso8859\_10* ISO 8859-10 with samii characters.  
*sosi\_charset\_nd7* Norsk Data 7-bit.  
*sosi\_charset\_utf8* UTF-8.

Definition at line 126 of file sosi\_types.h.

## **enum sosicon::sosi::ElementType**

List of SOSI element types.

### **Enumerator**

*sosi\_element\_unknown* Unknown element.  
*sosi\_element\_address\_identifier* Street address identifier.  
*sosi\_element\_airport\_roads* Airport roads.  
*sosi\_element\_airport\_type* Airport type.  
*sosi\_element\_area* Area.  
*sosi\_element\_charset* Character set.  
*sosi\_element\_coordsys* Grid type.  
*sosi\_element\_curve* Curve.  
*sosi\_element\_data\_collection\_date* Data collection date.  
*sosi\_element\_eof* End of file.  
*sosi\_element\_head* Header.  
*sosi\_element\_height* Height.  
*sosi\_element\_iata\_code* IATA code (aviation)  
*sosi\_element\_icao\_code* ICAO code (aviation)  
*sosi\_element\_kp* Junction point.  
*sosi\_element\_level* SOSI level.  
*sosi\_element\_max\_ne* Maximum north-east (bbox)  
*sosi\_element\_min\_ne* Minimum north-east (bbox)  
*sosi\_element\_municipality* Municipality.  
*sosi\_element\_name* Name.  
*sosi\_element\_ne* North-east coordinate.  
*sosi\_element\_neh* North-east/height coordinate.  
*sosi\_element\_objtype* Object type.  
*sosi\_element\_origo\_ne* Origo north-east.  
*sosi\_element\_owner* Dataset owner.

*sosi\_element\_point* Point.  
*sosi\_element\_quality* Quality of data.  
*sosi\_element\_ref* Element reference.  
*sosi\_element\_surface* Surface.  
*sosi\_element\_text* Text.  
*sosi\_element\_traffic\_type* Traffic type.  
*sosi\_element\_transpar* Datum/projection/coordinate system.  
*sosi\_element\_unit* Resolution (fraction of a metre)  
*sosi\_element\_updatedate* Update date.  
*sosi\_element\_water\_width* Water width.  
*sosi\_element\_vendor* Data vendor.  
*sosi\_element\_version* SOSI version.

Definition at line 38 of file sosi\_types.h.

#### **enum sosicon::sosi::JunctionPoint**

Default SOSI junction point layer types.

##### **Enumerator**

*sosi\_junction\_node* Node point (KP 1)  
*sosi\_junction\_connection* Connection point (KP 900)  
*sosi\_junction\_open\_end* Valid open-ended point (KP 999)

Definition at line 138 of file sosi\_types.h.

#### **enum sosicon::sosi::ObjType**

List of SOSI OBJTYPES.

##### **Enumerator**

*sosi\_objtype\_unknown* Unknown or no feature.  
*sosi\_objtype\_airport* Airport.  
*sosi\_objtype\_airport\_type* Airport type.  
*sosi\_objtype\_baseline* Baseline.  
*sosi\_objtype\_carriageway* Carriageway.  
*sosi\_objtype\_cadastral\_address* Cadastral address.  
*sosi\_objtype\_coastline* Coast line.  
*sosi\_objtype\_constituency\_boundary* Constituency boundary.  
*sosi\_objtype\_county\_boundary* County boundary.  
*sosi\_objtype\_data\_delineation* Clipping path.  
*sosi\_objtype\_edge\_view* Edge view.

*sosi\_objtype\_fictitious\_dividing\_line* Line splitting large surfeces.  
*sosi\_objtype\_forest* Forest.  
*sosi\_objtype\_developed\_area* Built up area.  
*sosi\_objtype\_golf\_course* Golf course.  
*sosi\_objtype\_industrial\_area* Industrial area.  
*sosi\_objtype\_lake* Lake.  
*sosi\_objtype\_lane* Driving lane.  
*sosi\_objtype\_lake\_edge* Lake edge.  
*sosi\_objtype\_lake\_river\_barrier* Lake-to-river delimitation.  
*sosi\_objtype\_land\_use\_boundary* Land use border.  
*sosi\_objtype\_level\_crossing* Track level crossing.  
*sosi\_objtype\_municipal\_divide* Municipal boundary crossing.  
*sosi\_objtype\_municipality* Municipality.  
*sosi\_objtype\_municipality\_boundary* Municipality boundary.  
*sosi\_objtype\_marsh* Marsh.  
*sosi\_objtype\_national\_border* National border.  
*sosi\_objtype\_pedestrian\_bicycle\_road\_centre\_line* mid-way line  
*sosi\_objtype\_sea\_river\_delineation* Sea or river delineation.  
*sosi\_objtype\_snow\_field* Snow/glacier.  
*sosi\_objtype\_open\_land* Open land.  
*sosi\_objtype\_river\_brook* River or stream.  
*sosi\_objtype\_river\_brook\_edge* River or stream bank.  
*sosi\_objtype\_road\_block* Road block.  
*sosi\_objtype\_road\_centre\_line* Road centre line.  
*sosi\_objtype\_road\_under\_railway* Road under railway.  
*sosi\_objtype\_sea\_surface* Sea surface.  
*sosi\_objtype\_sidewalk* Sidewalk.  
*sosi\_objtype\_spelling* Spelling of place name.  
*sosi\_objtype\_stone\_quarry* Area for stone quarry.  
*sosi\_objtype\_street\_address* Street address.  
*sosi\_objtype\_territorial\_boundary* Territorial boundary (nautical)  
*sosi\_objtype\_turn\_connecting\_segment* Turn connection segment (artificial)

Definition at line 79 of file sosi\_types.h.

## Function Documentation

### ElementType sosicon::sosi::sosiNameToType (std::string *sosiElementName*)

Convert SOSI element names to ElementType enum value.

The enum member names are translations of the Norwegian element names.

#### Parameters:

<i>std::string</i>	<i>sosiElementName</i> The standard SOSI element name in Norwegian.
--------------------	---

#### Returns:

ElementType enumeration value representing current element type.

### ObjType sosicon::sosi::sosiObjNameToType (std::string *sosiObjTypeName*)

Convert SOSI objtype names to ObjType enum value.

The enum member names are translations of the Norwegian geographic features.

#### Parameters:

<i>std::string</i>	<i>sosiObjtypeName</i> The standard SOSI objtype name in Norwegian.
--------------------	---

#### Returns:

ObjType enumeration value representing current element objtype.

### CoordSys sosicon::sosi::sysCodeToCoordSys (int *sysCode*)

Convert SOSI SYSKODE value to coordinate system data.

#### Parameters:

<i>int</i>	The SOSI SYSKODE value.
------------	-------------------------

#### Returns:

CoordSys structure with information about the requested coordinate system.

# Namespace Documentation

## sosicon Namespace Reference

Application root.

### Namespaces

- **byteOrder**
- *Big/low-endian conversions.* **shape**
- *ESRI Shape.* **sosi**
- *SOSI.* **utils**

### String manipulation routines. Classes

- class **CommandLine**
- *Command-line parser.* class **ConverterSosi2psql**
- *SOSI to PostgreSQL/PostGIS converter.* class **ConverterSosi2shp**
- *SOSI to ESRI Shape converter.* class **ConverterSosi2tsv**
- *SOSI to TSV converter.* class **ConverterSosi2xml**
- *SOSI to ESRI Shape converter.* class **ConverterSosiStat**
- *SOSI to ESRI Shape converter.* class **Coordinate**
- *Coordinate container.* class **CoordinateCollection**
- *Coordinate container.* class **EventDispatcher**
- *Event dispatcher template class.* class **Factory**
- *Factory class.* class **IBinaryStreamable**
- *Interface: Binary streamable object.* class **IConverter**
- *Interface: Converter.* class **ICoordinate**
- *Interface: Coordinate.* class **ILookupTable**
- *Interface: Lookup table.* class **IRectangle**
- *Interface: Rectangle.* class **IShapeElement**
- *Interface: Shape element.* class **IShapeElementHeader**
- *Interface: Shape element header.* class **IShapefile**
- *Interface: Shapefile.* class **IShapefileDbfPart**
- *Interface: ShapefileDbfPart.* class **IShapefilePrjPart**
- *Interface: ShapefilePrjPart.* class **IShapefileShpPart**
- *Interface: ShapefileShpPart.* class **IShapefileShxPart**
- *Interface: ShapefileShxPart.* class **IShapeHeader**
- *Interface: Shape element.* class **ISosiElement**
- *Interface: SOSI element.* class **ISosiHeadMember**
- *Interface: SOSI header element.* class **LogEvent**
- *Log event.* class **LogEventDispatcher**
- class **Logger**
- *SOSI logger.* class **Parser**

### SOSI file parser. Typedefs

- typedef std::vector< **ICoordinate** \* > **CoordinateList**  
*List of coordinate pairs.*

## Enumerations

- enum **Wkt** { **wkt\_unknown**, **wkt\_point**, **wkt\_linestring**, **wkt\_polygon** } *List of applied, well-known text geometries.*

## Functions

- bool **getNext** (**ICoordinate** \* &coord, **sosi::NorthEastList** &list, **sosi::NorthEastList::iterator** &i)  
*Get next coordinate in list.*
- bool **getNextOffset** (int &offset, std::vector< int > &offsets, std::vector< int >::iterator &iterator)  
*Get next offset in part offsets list.*
- bool **isClockwise** (std::vector< **ICoordinate** \* >::iterator &begin, std::vector< **ICoordinate** \* >::iterator &end)  
*Analyzes polygon direction.*
- bool **isCounterClockwise** (std::vector< **ICoordinate** \* >::iterator &begin, std::vector< **ICoordinate** \* >::iterator &end)  
*Analyzes polygon direction.*
- void **neListToCoordList** (**sosi::NorthEastList** &neList, std::vector< **ICoordinate** \* > &coordList)  
*Extracts single coordinates from list of North-East elements.*
- std::ostream & **operator<<** (std::ostream &os, **IBinaryStreamable** &binaryStreamable)  
*Stream output operator.*
- **Logger** & **flush** (**Logger** &l)

## Variables

- **Logger** **logstream**

---

## Detailed Description

Application root.

---

## Typedef Documentation

**typedef std::vector< ICoordinate \* > sosicon::CoordinateList**

List of coordinate pairs.

Used throughout the application.

Definition at line 30 of file common\_types.h.

---

## Enumeration Type Documentation

**enum sosicon::Wkt**

List of applied, well-known text geometries.

**Enumerator**

**wkt\_unknown** Unknown geometry.

**wkt\_point** Point geometry.



*wkt\_linestring* Linestring geometry.

*wkt\_polygon* Polygon geometry.

Definition at line 33 of file common\_types.h.

---

## Function Documentation

**sosicon::Logger & sosicon::flush (sosicon::Logger & *l*)**

Definition at line 85 of file logger.cpp.

**bool sosicon::getNext (ICoordinate \*& *coord*, sosi::NorthEastList & *list*,  
sosi::NorthEastList::iterator & *i*)**

Get next coordinate in list.

Definition at line 21 of file coordinate\_collection.cpp.

**bool sosicon::getNextOffset (int & *offset*, std::vector< int > & *offsets*, std::vector< int >::iterator  
& *iterator*)**

Get next offset in part offsets list.

Definition at line 42 of file coordinate\_collection.cpp.

**bool sosicon::isClockwise (std::vector< ICoordinate \* >::iterator & *begin*, std::vector<  
ICoordinate \* >::iterator & *end*)**

Analyzes polygon direction.

Checks a series of coordinates to see if they are ordered in a clockwise manner.

### Parameters:

<i>begin</i>	Iterator to the first item to be analyzed.
<i>end</i>	Iterator to the end item, one item past the last one to be analyzed.

### Returns:

true if the coordinates are ordered clockwise.

Definition at line 60 of file coordinate\_collection.cpp.

**bool sosicon::isCounterClockwise (std::vector< ICoordinate \* >::iterator & *begin*, std::vector<  
ICoordinate \* >::iterator & *end*)**

Analyzes polygon direction.

Checks a series of coordinates to see if they are ordered in a counter-clockwise manner.

### Parameters:

<i>begin</i>	Iterator to the first item to be analyzed.
<i>end</i>	Iterator to the end item, one item past the last one to be analyzed.

**Returns:**

true if the coordinates are ordered counter-clockwise.

Definition at line 55 of file coordinate\_collection.cpp.

**void sosicon::neListToCoordList (sosi::NorthEastList & *neList*, std::vector< ICoordinate \* > & *coordList*)**

Extracts single coordinates from list of North-East elements.

Converts a vector of NE elements to a vector of coordinates.

**Parameters:**

<i>neList</i>	The source vector.
<i>coordList</i>	The destination vector.

Definition at line 72 of file coordinate\_collection.cpp.

---

## Variable Documentation

**sosicon::Logger sosicon::logstream**

Definition at line 21 of file logger.cpp.

## sosicon::byteOrder Namespace Reference

Big/low-endian conversions.

### Enumerations

- enum **Endianness** { **not\_set**, **big**, **little** } *Big/little flag.*

### Functions

- **Endianness determine** ()  
*Determines system endianness.*
- void **doubleToLittleEndian** (double from, char \*to)  
*Writes little endian representation of double.*
- void **toBigEndian** (const char \*from, char \*to, size\_t bufSize)  
*Reverses buffer to big endian if required.*
- void **toLittleEndian** (const char \*from, char \*to, size\_t bufSize)  
*Reverses buffer to little endian if required.*

### Variables

- enum **sosicon::byteOrder::Endianness**                      **endianness**  
*Stores system endianness.*

---

## Detailed Description

Big/low-endian conversions.

---

## Enumeration Type Documentation

### enum sosicon::byteOrder::Endianness

Big/little flag.

#### Enumerator

*not\_set*  
*big*  
*little*

Definition at line 36 of file byte\_order.h.

---

## Function Documentation

### sosicon::byteOrder::Endianness sosicon::byteOrder::determine ()

Determines system endianness.

Tests byte-order to see if the program runs on a big endian or a little endian architecture. Flags the `byteOrder::endian` variable.

**Returns:**

System endianness.

**Return values:**

<i>Endianness::big</i>	Big endian system.
<i>Endianness::little</i>	Little endian system.

Definition at line 24 of file `byte_order.cpp`.

**void sosicon::byteOrder::doubleToLittleEndian (double *from*, char \* *to*)**

Writes little endian representation of double.

Serializes double-precision floating point value to IEEE little endian representation for binary embedding in files.

**Parameters:**

<i>from</i>	The double value to parse.
<i>to</i>	pointer to destination buffer. The buffer must be at least 8 bytes wide, as this is the size of the IEEE 754 format.

Definition at line 56 of file `byte_order.cpp`.

**void sosicon::byteOrder::toBigEndian (const char \* *from*, char \* *to*, size\_t *bufSize*)**

Reverses buffer to big endian if required.

Copies source buffer to destination buffer. If the program runs on a little-endian system, the byte order will be reversed.

**Parameters:**

<i>from</i>	pointer to source buffer.
<i>to</i>	pointer to destination buffer. The buffer must be at least as big as the source buffer.
<i>bufSize</i>	The number of bytes to copy.

Definition at line 36 of file `byte_order.cpp`.

**void sosicon::byteOrder::toLittleEndian (const char \* *from*, char \* *to*, size\_t *bufSize*)**

Reverses buffer to little endian if required.

Copies source buffer to destination buffer. If the program runs on a big-endian system, the byte order will be reversed.

**Parameters:**

<i>from</i>	pointer to source buffer.
<i>to</i>	pointer to destination buffer. The buffer must be at least as big as the source buffer.
<i>bufSize</i>	The number of bytes to copy.

Definition at line 46 of file `byte_order.cpp`.

## Variable Documentation

**enum sosicon::byteOrder::Endianness**

Stores system endianness.

**sosicon::byteOrder::endianness**

## sosicon::shape Namespace Reference

ESRI Shape.

### Classes

- union **DoubleField**
- *32 bit double / byte field* union **Int16Field**
- *16 bit integer / byte field* union **Int32Field**
- *32 bit integer / byte field* union **Int32TField**
- *32 bit integer / byte / geom::ShapeType field* union **Int8Field**
- *8 bit integer / byte field* class **Shapefile**
- *Shapefile implementation.* struct **ShxIndex**

### Typedefs

- typedef std::map< std::string, std::string > **DbfRecord**
- typedef std::vector< **DbfRecord** > **DbfRecordSet**
- typedef std::map< std::string, int > **DbfFieldLengths**
- typedef std::vector< **ShxIndex** > **ShxOffsets**

### Enumerations

- enum **ShapeType** { **shape\_type\_none**, **shape\_type\_nullShape**, **shape\_type\_point**, **shape\_type\_polyLine**, **shape\_type\_polygon**, **shape\_type\_multipart**, **shape\_type\_pointZ**, **shape\_type\_polyLineZ**, **shape\_type\_polygonZ**, **shape\_type\_multipartZ**, **shape\_type\_pointM**, **shape\_type\_polyLineM**, **shape\_type\_polygonM**, **shape\_type\_multiPointM**, **shape\_type\_multiPatch** } *Geometry types.*

### Functions

- **ShapeType** **getShapeEquivalent** (sosi::ElementType sosiType)  
*Resolve geometry type.*

---

## Detailed Description

ESRI Shape.

---

## Typedef Documentation

**typedef std::map<std::string, int> sosicon::shape::DbfFieldLengths**

Definition at line 90 of file shapefile\_types.h.

**typedef std::map<std::string, std::string> sosicon::shape::DbfRecord**

Definition at line 88 of file shapefile\_types.h.

**typedef std::vector<DbfRecord> sosicon::shape::DbfRecordSet**

Definition at line 89 of file shapefile\_types.h.

**typedef std::vector<ShxIndex> sosicon::shape::ShxOffsets**

Definition at line 91 of file shapefile\_types.h.

---

## Enumeration Type Documentation

**enum sosicon::shape::ShapeType**

Geometry types.

The numeric values are in accordance with the shapefile specification.

### Enumerator

*shape\_type\_none*  
*shape\_type\_nullShape*  
*shape\_type\_point*  
*shape\_type\_polyLine*  
*shape\_type\_polygon*  
*shape\_type\_multipoint*  
*shape\_type\_pointZ*  
*shape\_type\_polyLineZ*  
*shape\_type\_polygonZ*  
*shape\_type\_multipointZ*  
*shape\_type\_pointM*  
*shape\_type\_polyLineM*  
*shape\_type\_polygonM*  
*shape\_type\_multiPointM*  
*shape\_type\_multiPatch*

Definition at line 34 of file shapefile\_types.h.

---

## Function Documentation

**sosicon::shape::ShapeType sosicon::shape::getShapeEquivalent (sosi::ElementType *sosiType*)**

Resolve geometry type.

Translate SOSI geomtry type to corresponding shape geometry, if applicable

Definition at line 21 of file shapefile.cpp.

## sosicon::sosi Namespace Reference

SOSI.

### Namespaces

- **chartables**

### Classes

- class **CoordSys**
- *SOSI coordinate system.* struct **ReferenceData**
- *SOSI reference number.* class **SosiCharsetSingleton**
- *SOSI Character set.* class **SosiElement**
- *Basic SOSI element.* class **SosiElementSearch**
- class **SosiJunctionPoint**
- *SOSI Junction point.* class **SosiNorthEast**
- *SOSI North-east element.* class **SosiOrigoNE**
- *SOSI Junction point.* class **SosiRefList**
- *SOSI REF list.* class **SosiTranslationTable**
- class **SosiUnit**

### SOSI Unit. Typedefs

- typedef std::map< std::string, **ISosiElement** \* > **SosiElementMap**  
*Element index type.*
- typedef std::vector< **ISosiElement** \* > **SosiChildrenList**
- typedef SosiChildrenList::iterator **SosiChildrenIterator**
- typedef std::vector< **SosiNorthEast** \* > **NorthEastList**  
*List of SosiNorthEast elements.*
- typedef std::vector< **ReferenceData** \* > **GeometryRef**  
*List of SOSI references.*
- typedef std::vector< **GeometryRef** \* > **GeometryCollection**  
*Collection of SOSI reference lists.*

### Enumerations

- enum **ElementType** { **sosi\_element\_unknown**, **sosi\_element\_address\_identifier**, **sosi\_element\_airport\_roads**, **sosi\_element\_airport\_type**, **sosi\_element\_area**, **sosi\_element\_charset**, **sosi\_element\_coordsys**, **sosi\_element\_curve**, **sosi\_element\_data\_collection\_date**, **sosi\_element\_eof**, **sosi\_element\_head**, **sosi\_element\_height**, **sosi\_element\_iata\_code**, **sosi\_element\_icao\_code**, **sosi\_element\_kp**, **sosi\_element\_level**, **sosi\_element\_max\_ne**, **sosi\_element\_min\_ne**, **sosi\_element\_municipality**, **sosi\_element\_name**, **sosi\_element\_ne**, **sosi\_element\_neh**, **sosi\_element\_objtype**, **sosi\_element\_origo\_ne**, **sosi\_element\_owner**, **sosi\_element\_point**, **sosi\_element\_quality**, **sosi\_element\_ref**, **sosi\_element\_surface**, **sosi\_element\_text**, **sosi\_element\_traffic\_type**, **sosi\_element\_transpar**, **sosi\_element\_unit**, **sosi\_element\_updatedate**, **sosi\_element\_water\_width**, **sosi\_element\_vendor**, **sosi\_element\_version** } *List of SOSI element types.*
- enum **ObjType** { **sosi\_objtype\_unknown**, **sosi\_objtype\_airport**, **sosi\_objtype\_airport\_type**, **sosi\_objtype\_baseline**, **sosi\_objtype\_carriageway**, **sosi\_objtype\_cadastral\_address**, **sosi\_objtype\_coastline**, **sosi\_objtype\_constituency\_boundary**, **sosi\_objtype\_county\_boundary**, **sosi\_objtype\_data\_delineation**, **sosi\_objtype\_edge\_view**, **sosi\_objtype\_fictious\_dividing\_line**, **sosi\_objtype\_forest**, **sosi\_objtype\_developed\_area**, **sosi\_objtype\_golf\_course**, **sosi\_objtype\_industrial\_area**, **sosi\_objtype\_lake**, **sosi\_objtype\_lane**, **sosi\_objtype\_lake\_edge**, **sosi\_objtype\_lake\_river\_barrier**,



- sosi\_objtype\_land\_use\_boundary, sosi\_objtype\_level\_crossing, sosi\_objtype\_municipal\_divide, sosi\_objtype\_municipality, sosi\_objtype\_municipality\_boundary, sosi\_objtype\_marsh, sosi\_objtype\_national\_border, sosi\_objtype\_pedestrian\_bicycle\_road\_centre\_line, sosi\_objtype\_sea\_river\_delineation, sosi\_objtype\_snow\_field, sosi\_objtype\_open\_land, sosi\_objtype\_river\_brook, sosi\_objtype\_river\_brook\_edge, sosi\_objtype\_road\_block, sosi\_objtype\_road\_centre\_line, sosi\_objtype\_road\_under\_railway, sosi\_objtype\_sea\_surface, sosi\_objtype\_sidewalk, sosi\_objtype\_spelling, sosi\_objtype\_stone\_quarry, sosi\_objtype\_street\_address, sosi\_objtype\_territorial\_boundary, sosi\_objtype\_turn\_connecting\_segment }*List of SOSI OBJTYPEs.*
- enum **Charset** { sosi\_charset\_undetermined, sosi\_charset\_ansi, sosi\_charset\_decn7, sosi\_charset\_dosn8, sosi\_charset\_iso8859\_1, sosi\_charset\_iso8859\_10, sosi\_charset\_nd7, sosi\_charset\_utf8 }*SOSI character encodings.*
- enum **JunctionPoint** { sosi\_junction\_node, sosi\_junction\_connection, sosi\_junction\_open\_end }*Default SOSI junction point layer types.*

## Functions

- **CoordSys sysCodeToCoordSys** (int sysCode)  
*Convert SOSI SYSKODE value to coordinate system data.*
- **ElementType sosiNameToType** (std::string sosiElementName)  
*Convert SOSI element names to ElementType enum value.*
- **ObjType sosiObjNameToType** (std::string sosiObjTypeName)  
*Convert SOSI objtype names to ObjType enum value.*
- void **deleteNorthEast**s (NorthEastList &lst)  
*Deletes SosiNorthEast elements of NorthEastList.*

---

## Detailed Description

SOSI.

---

## Typedef Documentation

**typedef std::vector<SosiNorthEast\*> sosicon::sosi::NorthEastList**

List of SosiNorthEast elements.

Definition at line 115 of file sosi\_north\_east.h.

---

## Function Documentation

**void sosicon::sosi::deleteNorthEast**s (NorthEastList & lst)

Deletes **SosiNorthEast** elements of NorthEastList.

Definition at line 21 of file sosi\_north\_east.cpp.

## **sosicon::sosi::chartables Namespace Reference**

## sosicon::utils Namespace Reference

String manipulation routines.

### Functions

- `std::string className2FileName (const std::string &className)`  
*Converts Class name to file name string.*
- `std::vector< std::string > explode (char delimiter, std::string str)`  
*Split a string by a character.*
- `bool fileExists (const std::string &name)`  
*Test if file exists.*
- `bool isNumeric (const std::string &str)`  
*Test if a string represents a numeric value.*
- `std::string nonExistingFilename (std::string defaultName)`  
*Asserts output file name to be non-existing.*
- `std::string normalizeAppClassName (const std::string &className)`  
*Asserts correct name of application classes.*
- `std::string purgeCrLf (std::string str)`  
*Remove carriage returns and line feeds.*
- `std::string repeat (const std::string &seq, unsigned int count)`  
*Repeat string N times.*
- `std::string replaceAll (const std::string &from, const std::string &to, const std::string &subject)`  
*Replace all occurrences of one string with another.*
- `std::string sqlNormalize (const std::string &str)`  
*Sanitizes SQL data string.*
- `std::string stripTrailingSlash (const std::string &str)`  
*Remove trailing forward- and backward slashes from path component.*
- `std::string trim (const std::string &str)`  
*Removes leading and trailing space characters.*
- `std::string trimLeft (const std::string &str)`
- `std::string trimRight (const std::string &str)`
- `std::string toFieldname (const std::string &from)`  
*Substitutes Norwegian characters.*
- `std::string toLower (const std::string &from)`
- `std::string ucFirst (const std::string &str)`
- `std::string unquote (const std::string &str)`  
*Remove quotes around string.*
- `void getPathInfo (std::string path, std::string &dir, std::string &tit, std::string &ext)`
- `std::string wktToStr (Wkt wktGeom)`  
*Get Well Known Text from Wkt enum.*

---

### Detailed Description

String manipulation routines.

---

## Function Documentation

### **string sosicon::utils::className2FileName (const std::string & *className*)**

Converts Class name to file name string.

Class names are written in pascal case (i.e. 'CarmineEntity', 'XMLParser'). This method constructs a file name string for a given class name (i.e. 'carmine\_entity', 'xml\_parser'). The file names are always written in lower case, with underscores separating the words.

#### **Parameters:**

<i>className</i>	The pascal-cased class name to convert to a file name.
------------------	--

#### **Returns:**

The file name string without extension.

Definition at line 23 of file utils.cpp.

### **std::vector< std::string > sosicon::utils::explode (char *delimiter*, std::string *str*)**

Split a string by a character.

The source string *str* is split by the *delimiter* character, and each part is put sequentially in a vector of strings, excluding the *delimiter* character.

#### **Parameters:**

<i>delimiter</i>	The delimiter character, typically a comma or a semicolon.
<i>str</i>	The source string to be split into a vector of substrings.

#### **Returns:**

A vector of strings, each of which are substrings of *str*.

Definition at line 44 of file utils.cpp.

### **bool sosicon::utils::fileExists (const std::string & *name*) [inline]**

Test if file exists.

Definition at line 61 of file utils.h.

### **void sosicon::utils::getPathInfo (std::string *path*, std::string & *dir*, std::string & *tit*, std::string & *ext*)**

Definition at line 296 of file utils.cpp.

### **bool sosicon::utils::isNumeric (const std::string & *str*)**

Test if a string represents a numeric value.

Returns true if the provided string contains numbers only, and if the first digit is not zero. Numbers with leading zeros should be treated as strings, since they might be phone numbers, post numbers or municipal codes.

#### **Parameters:**

<i>str</i>	The string value to test.
------------	---------------------------

**Returns:**

True if the string represents a numeric value with no leading zero, otherwise false.  
Definition at line 66 of file utils.cpp.

**std::string sosicon::utils::nonExistingFilename (std::string *defaultName*)**

Asserts output file name to be non-existing.

Tests candidate file paths to find a unique output file name, appending and incrementing a serial number until a "free" name is encountered.

**Parameters:**

<i>defaultName</i>	If the output file name is not specified on the command-line, the default file name will be used as a starting point.
--------------------	---

**Returns:**

Path to non-existing output file.  
Definition at line 82 of file utils.cpp.

**string sosicon::utils::normalizeAppClassName (const std::string & *className*)**

Asserts correct name of application classes.

Application classes should always begin with the 'App' prefix. This method adds the prefix to the provided class name if it is missing.

**Parameters:**

<i>className</i>	The class name string to be resolved and normalized.
------------------	--

**Returns:**

Normalized and corrected class name string.  
Definition at line 103 of file utils.cpp.

**std::string sosicon::utils::purgeCrLf (std::string *str*)**

Remove carriage returns and line feeds.

Removes newlines from the target string. The or  
characters may be anywhere in the string.

**Parameters:**

<i>str</i>	The string to be purged.
------------	--------------------------

**Returns:**

The result string.

**string sosicon::utils::repeat (const std::string & *seq*, unsigned int *count*)**

Repeat string N times.

Creates a new string containing the provided string sequence for a predetermined number of repetitions.

**Parameters:**

<i>seq</i>	Reference to the string to be repeated.
------------	---

<i>count</i>	The numner of times to repeat the string sequence.
--------------	--

**Returns:**

The result string.

Definition at line 130 of file utils.cpp.

**string sosicon::utils::replaceAll (const std::string & *from*, const std::string & *to*, const std::string & *subject*)**

Replace all occurences of one string with another.

Searches for a given string sequence, replacing all occurences by th provided substitution string.

**Parameters:**

<i>from</i>	The string sequence to be changed.
<i>to</i>	The string to replace the 'from' sequence with.
<i>subject</i>	The string to perform the search on.

**Returns:**

The new string, a copy of 'subject' where all occurences of 'from' are replaced with 'to'.

Definition at line 141 of file utils.cpp.

**string sosicon::utils::sqlNormalize (const std::string & *str*)**

Sanitizes SQL data string.

Escapes special characters in a string for use in an SQL statement.

**Parameters:**

<i>str</i>	The target string.
------------	--------------------

**Returns:**

A copy of the target string, with reserved characters escaped.

Definition at line 155 of file utils.cpp.

**string sosicon::utils::stripTrailingSlash (const std::string & *str*)**

Remove trailing forward- and backward slashes from path component.

Definition at line 267 of file utils.cpp.

**string sosicon::utils::toFieldName (const std::string & *from*)**

Substitutes Norwegian characters.

Definition at line 179 of file utils.cpp.

**string sosicon::utils::toLower (const std::string & *from*)**

Definition at line 209 of file utils.cpp.

**string sosicon::utils::trim (const std::string & *str*)**

Removes leading and trailing space characters.

Space characters in the beginning and at the end of the source string are trimmed.

**Parameters:**

<i>str</i>	The target string.
------------	--------------------

**Returns:**

A copy of the target string, without leading and/or trailing space characters.

Definition at line 226 of file utils.cpp.

**string sosicon::utils::trimLeft (const std::string & *str*)**

Definition at line 232 of file utils.cpp.

**string sosicon::utils::trimRight (const std::string & *str*)**

Definition at line 239 of file utils.cpp.

**string sosicon::utils::ucFirst (const std::string & *str*)**

Definition at line 247 of file utils.cpp.

**string sosicon::utils::unquote (const std::string & *str*)**

Remove quotes around string.

Definition at line 280 of file utils.cpp.

**std::string sosicon::utils::wktToStr (Wkt *wktGeom*)**

Get Well Known Text from Wkt enum.

Definition at line 335 of file utils.cpp.

# Class Documentation

## sosicon::CommandLine Class Reference

Command-line parser.

```
#include <command_line.h>
```

### Public Member Functions

- void **outputHelpText** ()  
*Display help text.*
- void **outputDisclaimer** ()  
*Display disclaimer.*
- void **outputLicense** ()  
*Display license.*
- void **parse** (int argc, char \*argv[])  
*Read command-line arguments.*
- void **parse** (std::string cmdStr)  
*Read command-line string.*
- **CommandLine** ()  
*Constructor.*
- virtual **~CommandLine** ()  
*Destructor.*

### Public Attributes

- std::string **mCommand**  
*Conversion command.*
- bool **mCreateStatements**  
*Build create statements only.*
- bool **mInsertStatements**  
*Build insert statements only.*
- std::vector< std::string > **mSourceFiles**  
*List of input files.*
- std::vector< std::string > **mObjTypes**  
*List of object types to output.*
- std::vector< std::string > **mFilterSosiId**  
*Export specific SOSI elements.*
- std::vector< std::string > **mGeomTypes**  
*List of geometry types to output.*
- std::vector< std::string > **mFieldSelection**  
*List of selected fields.*
- std::string **mDestinationDirectory**  
*Destination directory.*
- std::string **mOutputFile**  
*Destination file.*



- **bool mIsTtyIn**  
*TTY in flag.*
- **bool mIsTtyOut**  
*TTY out flag.*
- **bool mAppend**  
*Append flag.*
- **std::string mDbSchema**  
*PostgreSQL database schema.*
- **std::string mDbTable**  
*PostgreSQL database table.*
- **bool mIncludeHeader**  
*Include column headers.*
- **bool mMakeSubDir**  
*Create a sub directory for the output files.*
- **std::string mSrid**  
*Specifies SRID for exports.*
- **int mVerbose**  
*Verbose output.*

---

## Detailed Description

Command-line parser.

### Author:

Espen Andersen

### Copyright:

GNU General Public License

Takes the arguments from the command-line and parses them into the class member variables. On Linux systems, this class also reads piped content (file list to be processed) from stdin, using it as input parameters.

Definition at line 51 of file `command_line.h`.

---

## Constructor & Destructor Documentation

### **sosicon::CommandLine::CommandLine ()**

Constructor.

Definition at line 21 of file `command_line.cpp`.

### **sosicon::CommandLine::~~CommandLine () [virtual]**

Destructor.

Definition at line 48 of file `command_line.cpp`.

---

## Member Function Documentation

### **void sosicon::CommandLine::outputDisclaimer ()**

Display disclaimer.

Outputs disclaimer text.

Definition at line 283 of file `command_line.cpp`.

### **void sosicon::CommandLine::outputHelpText ()**

Display help text.

Outputs simple help text to the command-line.

Definition at line 214 of file `command_line.cpp`.

### **void sosicon::CommandLine::outputLicense ()**

Display license.

Outputs lisence text.

Definition at line 320 of file `command_line.cpp`.

### **void sosicon::CommandLine::parse (int *argc*, char \* *argv*[])**

Read command-line arguments.

Parses the command-line arguments and loads the settings into the member variables. This function will also read piped content (file name list) from stdin on linux systems, adding it to the **CommandLine::mSourceFiles** list of files to be processed.

#### **Parameters:**

<i>argc</i>	Number of arguments present. Passed on from <b>main()</b> function.
<i>argv</i>	Array of string pointers to each argument. Passed on from <b>main()</b> function.

Either or both, but not none (!)

Definition at line 95 of file `command_line.cpp`.

### **void sosicon::CommandLine::parse (std::string *cmdStr*)**

Read command-line string.

Parses the command-line string and loads the settings into the member variables.

#### **Parameters:**

<i>cmdStr</i>	Complete command-line string to be parsed.
---------------	--

Definition at line 63 of file `command_line.cpp`.

---

## Member Data Documentation

### **bool sosicon::CommandLine::mAppend**

Append flag.

If the destination file (-o ...) is specified together with the -a argument, this flag will be true to signal that data from several source files should be merged into one destination file.

Definition at line 153 of file command\_line.h.

### **std::string sosicon::CommandLine::mCommand**

Conversion command.

Specifies what type of conversion to perform. If this string is -2tsv, the SOSI file will be exported as tab separated values. The factory class uses this parameter to determine which **IConverter** implementation to employ upon initialization.

Definition at line 60 of file command\_line.h.

### **bool sosicon::CommandLine::mCreateStatements**

Build create statements only.

For PostgreSQL export: If this flag is set (by specifying the -create parameter), database table creation script will be output.

#### **Note:**

If neither -create nor -insert is specified, both create and insert statements are included in the export.

Definition at line 70 of file command\_line.h.

### **std::string sosicon::CommandLine::mDbSchema**

PostgreSQL database schema.

Name of database schema to export SOSI data to, when using -2psql converter.

Definition at line 159 of file command\_line.h.

### **std::string sosicon::CommandLine::mDbTable**

PostgreSQL database table.

Name of database table to export SOSI data to, when using -2psql converter.

Definition at line 165 of file command\_line.h.

### **std::string sosicon::CommandLine::mDestinationDirectory**

Destination directory.

Path to the target directory where the output files will be written. Specified by the -d argument.

Definition at line 127 of file command\_line.h.

### **std::vector<std::string> sosicon::CommandLine::mFieldSelection**

List of selected fields.

String vector containing the identifiers for the SOSI fields to be included in the export. Specified as a comma-separated list of strings following the -f argument.

Definition at line 120 of file command\_line.h.

### **std::vector<std::string> sosicon::CommandLine::mFilterSosild**

Export specific SOSI elements.

List of SOSI IDs of individual element/features to be exported. Specified as a comma-separated list of strings following the -id argument.

Definition at line 104 of file command\_line.h.

### **std::vector<std::string> sosicon::CommandLine::mGeomTypes**

List of geometry types to output.

String vector containing the geometry types for the elements to be included in the export. Relevant for shapefile exports, since shapefiles can only contain one geometry type at a time. Specified as a comma-separated list of strings following the -g argument. The converter will output one shapefile for each selected geometry.

Definition at line 113 of file command\_line.h.

### **bool sosicon::CommandLine::mIncludeHeader**

Include column headers.

For some output formats, such as tsv, this flag governs whether a line with the column header names should be included in the target file.

Definition at line 172 of file command\_line.h.

### **bool sosicon::CommandLine::mInsertStatements**

Build insert statements only.

For PostgreSQL export: If this flag is set (by specifying the -insert parameter), database table insertion script will be output.

#### **Note:**

If neither -create nor -insert is specified, both create and insert statements are included in the export.

Definition at line 80 of file command\_line.h.

### **bool sosicon::CommandLine::mIsTtyIn**

TTY in flag.

This flag is false if input is redirected (not a terminal window).

Definition at line 139 of file `command_line.h`.

#### **`bool sosicon::CommandLine::mIsTtyOut`**

TTY out flag.

This flag is false if output is redirected (not a terminal window).

Definition at line 145 of file `command_line.h`.

#### **`bool sosicon::CommandLine::mMakeSubDir`**

Create a sub directory for the output files.

If the `/s` switch is specified, this flag is set to true. Instead of emitting the output files directly to current directory, a sub directory will be created, to which the output files are written.

Definition at line 180 of file `command_line.h`.

#### **`std::vector<std::string> sosicon::CommandLine::mObjTypes`**

List of object types to output.

String vector containing the SOSI OBJTYPE identifiers for the elements to be included in the export. Specified as a comma-separated list of strings following the `-t` argument.

Definition at line 97 of file `command_line.h`.

#### **`std::string sosicon::CommandLine::mOutputFile`**

Destination file.

Specified by the `-o` argument. The target file name.

Definition at line 133 of file `command_line.h`.

#### **`std::vector<std::string> sosicon::CommandLine::mSourceFiles`**

List of input files.

String vector containing the list of SOSI input files to be converted. This list is populated either by the file names specified directly on the command-line, or by the content of stdin as piped in from other commands (such as `ls *.sos | ...`) on Linux based systems.

Definition at line 89 of file `command_line.h`.

#### **`std::string sosicon::CommandLine::mSrid`**

Specifies SRID for exports.

Used for grid conversion exports to postGIS or other conversions that supports this.

Definition at line 186 of file `command_line.h`.

**int sosicon::CommandLine::mVerbose**

Verbose output.

Verbose level. If this value is 0, no informative output will be emitted during file parsing. If the value is 1 (-v), limited output will be written to stdout - mostly file header information from each SOSI file to be converted. If the value is 2 (-V), a more comprehensive summary of every SOSI element in all source files will be output.

Definition at line 195 of file command\_line.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**command\_line.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**command\_line.cpp**

## sosicon::ConverterSosi2psql Class Reference

SOSI to PostgreSQL/PostGIS converter.

```
#include <converter_sosi2psql.h>
```

Inheritance diagram for sosicon::ConverterSosi2psql:



### Classes

- class **Field**

### Public Member Functions

- **ConverterSosi2psql** ()  
*Constructor.*
- virtual void **init** (**CommandLine** \*cmd)  
*Initialize converter.*
- virtual void **run** (bool \*cancel=0x00)  
*Start conversion.*

### Private Types

- typedef std::map< std::string, **Field** > **FieldsList**
- typedef std::map< **Wkt**, **FieldsList** \* > **FieldsListCollection**
- typedef std::vector< std::map< std::string, std::string > \* > **RowsList**
- typedef std::map< **Wkt**, **RowsList** \* > **RowsListCollection**

### Private Member Functions

- std::string **buildInsertStatements** (std::string dbSchema, std::string dbTable)  
*Build SQL insert statements for all geometries.*
- std::string **buildInsertStatement** (**Wkt** wktGeom, std::string dbSchema, std::string dbTable)  
*Build SQL insert statement for one geometry.*
- std::string **buildCreateStatements** (std::string sridDest, std::string dbSchema, std::string dbTable)  
*Build SQL create statements for all geometries.*
- std::string **buildCreateStatement** (**Wkt** wktGeom, std::string sridDest, std::string dbSchema, std::string dbTable)  
*Build SQL create statements for one geometry.*
- void **cleanup** ()
- void **cleanup** (**Wkt** wktGeom)
- void **extractData** (**ISosiElement** \*parent, **FieldsList** &hdr, std::map< std::string, std::string > \*&row)  
*Fetch element data fields recursively.*
- std::string **getSrid** (**ISosiElement** \*sosiTree)  
*Read current coordinate system from SOSI tree.*
- void **insertLineString** (**ISosiElement** \*lineString, std::string sridSource, std::string sridDest, std::string geomField)  
*Convert curve geomery (sosi KURVE) to SQL export data.*

- void **insertPoint** (**ISosiElement** \*point, std::string sridSource, std::string sridDest, std::string geomField)  
*Convert single point geomery (sosi PUNKT) to SQL export data.*
- void **insertPolygon** (**ISosiElement** \*polygon, std::string sridSource, std::string sridDest, std::string geomField)  
*Convert polygons (sosi FLATE) to SQL export data.*
- void **makePsql** (**ISosiElement** \*sosiTree, std::string sridDest, std::string dbSchema, std::string dbTable)  
*Make SQL dump from SOSI tree.*
- bool **objTypeExcluded** (sosi::SosiElementSearch &src)  
*Test if current element is filtered out by -t parameter.*
- void **writePsql** (std::string sridDest, std::string dbSchema, std::string dbTable)  
*Write SQL content.*
- virtual ~**ConverterSosi2psql** ()  
*Destructor.*

## Private Attributes

- **CommandLine** \* mCmd  
*Command line wrapper.*
- std::string **mCurrentSourcefile**  
*Souce file currently in process.*
- **FieldsListCollection** mFieldsListCollection  
*Collection of fields, one item for each geometry type.*
- **RowsListCollection** mRowsListCollection  
*Collection of rows, one item for each geometry type.*

---

## Detailed Description

SOSI to PostgreSQL/PostGIS converter.

If command-line parameter -2psql is specified, this converter will handle the output generation. Produces a PostgreSQL/PostGIS dump file from the SOSI source(s).

Definition at line 52 of file converter\_sosi2psql.h.

---

## Member Typedef Documentation

**typedef std::map< std::string,Field > sosicon::ConverterSosi2psql::FieldsList** [private]

Definition at line 92 of file converter\_sosi2psql.h.

**typedef std::map< Wkt, FieldsList\* > sosicon::ConverterSosi2psql::FieldsListCollection** [private]

Definition at line 93 of file converter\_sosi2psql.h.

**typedef std::vector< std::map< std::string,std::string >\* > sosicon::ConverterSosi2psql::RowsList** [private]

Definition at line 94 of file converter\_sosi2psql.h.



```
typedef std::map< Wkt, RowsList* > sosicon::ConverterSosi2psql::RowsListCollection [private]
```

Definition at line 95 of file converter\_sosi2psql.h.

---

## Constructor & Destructor Documentation

```
virtual sosicon::ConverterSosi2psql::~~ConverterSosi2psql () [inline], [private], [virtual]
```

Destructor.

Definition at line 307 of file converter\_sosi2psql.h.

```
sosicon::ConverterSosi2psql::ConverterSosi2psql () [inline]
```

Constructor.

Definition at line 312 of file converter\_sosi2psql.h.

---

## Member Function Documentation

```
std::string sosicon::ConverterSosi2psql::buildCreateStatement (Wkt   wktGeom, std::string  
sridDest, std::string  dbSchema, std::string  dbTable) [private]
```

Build SQL create statements for one geometry.

This function calls **sosicon::ConverterSosi2psql::buildCreateStatements** for each of the WKT geometries types to export.

### Parameters:

<i>dbSchema</i>	String representing the name of the database schema.
<i>dbTable</i>	String representing the base name of the database table. The name of the geometry for that table will be prepended to the base name.

### See also:

**sosicon::ConverterSosi2psql::buildCreateStatements()**

### Returns:

The SQL/DDDL creation script content.

Definition at line 44 of file converter\_sosi2psql.cpp.

```
std::string sosicon::ConverterSosi2psql::buildCreateStatements (std::string  sridDest, std::string  
dbSchema, std::string  dbTable) [private]
```

Build SQL create statements for all geometries.

This function calls **sosicon::ConverterSosi2psql::buildCreateStatement** for each of the WKT geometries types to export.

### Parameters:

<i>dbSchema</i>	String representing the name of the database schema.
-----------------	--

<i>dbTable</i>	String representing the base name of the database table. The name of the geometry for that table will be prepended to the base name.
----------------	--

**See also:**

**sosicon::ConverterSosi2psql::buildCreateStatement()**

**Returns:**

The SQL/DDDL creation script content.

Definition at line 21 of file converter\_sosi2psql.cpp.

**std::string sosicon::ConverterSosi2psql::buildInsertStatement (Wkt *wktGeom*, std::string *dbSchema*, std::string *dbTable*)[private]**

Build SQL insert statement for one geometry.

Creates the SQL statements required to insert the data for one WKT geometry.

**Parameters:**

<i>wktGeom</i>	WKT geometry type for current insertion script.
<i>dbSchema</i>	String representing the name of the database schema.
<i>dbTable</i>	String representing the base name of the database table. The name of the geometry for that table will be prepended to the base name.

**See also:**

**sosicon::ConverterSosi2psql::buildInsertStatements()**

**Returns:**

The SQL insertion script content.

Definition at line 146 of file converter\_sosi2psql.cpp.

**std::string sosicon::ConverterSosi2psql::buildInsertStatements (std::string *dbSchema*, std::string *dbTable*)[private]**

Build SQL insert statements for all geometries.

This function calls **sosicon::ConverterSosi2psql::buildInsertStatement** for each of the WKT geometries types to export.

**Parameters:**

<i>dbSchema</i>	String representing the name of the database schema.
<i>dbTable</i>	String representing the base name of the database table. The name of the geometry for that table will be prepended to the base name.

**See also:**

**sosicon::ConverterSosi2psql::buildInsertStatement()**

**Returns:**

The SQL insertion script content.

Definition at line 127 of file converter\_sosi2psql.cpp.

**void sosicon::ConverterSosi2psql::cleanup () [private]**

Release memory reserved for this converter. Called before destroying object. It may not be necessary to invoke **cleanup()** if the program is about to terminate anyway, as this can be quite time consuming after converting large files.

**See also:**

**sosicon::ConverterSosi2psql::cleanup( Wkt )**

Definition at line 229 of file converter\_sosi2psql.cpp.

**void sosicon::ConverterSosi2psql::cleanup( Wkt wktGeom ) [private]**

Release memory reserved for one geometry. Called from **sosicon::ConverterSosi2psql::cleanup( Wkt )**

**Parameters:**

<i>wktGeom</i>	The WKT geometry type for which to delete allocated memory.
----------------	---

**See also:**

**sosicon::ConverterSosi2psql::cleanup()**

Definition at line 237 of file converter\_sosi2psql.cpp.

**void sosicon::ConverterSosi2psql::extractData( ISosiElement \* parent, FieldsList & hdr, std::map< std::string, std::string > \* & row ) [private]**

Fetch element data fields recursively.

Traverses the SOSI element tree recursively, extracting plain data fields. The field names are stored in the *hdr* list, and the data values are stored in the *row* list. The data size value associated with each entry in *hdr* is updated to reflect the longest encountered field length.

**Parameters:**

<i>parent</i>	The SOSI (sub)tree to be traversed.
<i>hdr</i>	The fields list (table header).
<i>row</i>	The record set (table row).

Definition at line 254 of file converter\_sosi2psql.cpp.

**std::string sosicon::ConverterSosi2psql::getSrid( ISosiElement \* sosiTree ) [private]**

Read current coordinate system from SOSI tree.

Scans the SOSI header, searching for the KOORDSYS element with information about the current coordinate system.

**Parameters:**

<i>sosiTree</i>	Pointer to the root SOSI element.
-----------------	-----------------------------------

**Returns:**

The SRID code for the grid used in current file.

Definition at line 286 of file converter\_sosi2psql.cpp.

**virtual void sosicon::ConverterSosi2psql::init( CommandLine \* cmd ) [inline], [virtual]**

Initialize converter.

Implementation details in **sosicon::IConverter::init()**

**Parameters:**

<i>cmd</i>	Pointer to (the one and only) <b>CommandLine</b> instance.
------------	--

**See also:**

**sosicon::IConverter::init()**

Implements **sosicon::IConverter** (p.82).

Definition at line 320 of file converter\_sosi2psql.h.

**void sosicon::ConverterSosi2psql::insertLineString (ISosiElement \* *lineString*, std::string *sridSource*, std::string *sridDest*, std::string *geomField*) [private]**

Convert curve geomery (sosi KURVE) to SQL export data.

Extracts the coordinates from the given SOSI element and builds a linestring WKT geometry from it.

**See also:**

**sosicon::ConverterSosi2psql::insertPoint()**

**sosicon::ConverterSosi2psql::insertPolygon()**

**Parameters:**

<i>lineString</i>	SOSI geometry element (typically "KURVE").
<i>sridSource</i>	Spatial reference grid ID for the source file.
<i>sridDest</i>	Spatial reference grid ID for the target file.
<i>geomField</i>	The name of the field within the recordset representing the geometry data.

Definition at line 375 of file converter\_sosi2psql.cpp.

**void sosicon::ConverterSosi2psql::insertPoint (ISosiElement \* *point*, std::string *sridSource*, std::string *sridDest*, std::string *geomField*) [private]**

Convert single point geomery (sosi PUNKT) to SQL export data.

Extracts the coordinate from the given SOSI element and builds a point WKT geometry from it.

**See also:**

**sosicon::ConverterSosi2psql::insertLineString()**

**sosicon::ConverterSosi2psql::insertPolygon()**

**Parameters:**

<i>point</i>	SOSI geometry element (typically "PUNKT" or "TEKST").
<i>sridSource</i>	Spatial reference grid ID for the source file.
<i>sridDest</i>	Spatial reference grid ID for the target file.
<i>geomField</i>	The name of the field within the recordset representing the geometry data.

Definition at line 326 of file converter\_sosi2psql.cpp.

**void sosicon::ConverterSosi2psql::insertPolygon (ISosiElement \* *polygon*, std::string *sridSource*, std::string *sridDest*, std::string *geomField*) [private]**

Convert polygons (sosi FLATE) to SQL export data.

Extracts the coordinates from the given SOSI element and builds a polygon WKT geometry from it.

**See also:**

**sosicon::ConverterSosi2psql::insertLineString()**

**sosicon::ConverterSosi2psql::insertPoint()**

**Parameters:**

<i>point</i>	SOSI geometry element (typically "FLATE").
<i>sridSource</i>	Spatial reference grid ID for the source file.

<i>sridDest</i>	Spatial reference grid ID for the target file.
<i>geomField</i>	The name of the field within the recordset representing the geometry data.

Definition at line 428 of file converter\_sosi2psql.cpp.

```
void sosicon::ConverterSosi2psql::makePsql (ISosiElement *   sosiTree, std::string   sridDest,
std::string   dbSchema, std::string   dbTable)[private]
```

Make SQL dump from SOSI tree.

Iterates the SOSI tree once for each geometry to be exported, extracting relevant elements and passing them on to one of the insertion routines.

**Parameters:**

<i>sosiTree</i>	Pointer to the root SOSI element.
<i>sridDest</i>	Spatial reference grid ID for the target file.
<i>dbSchema</i>	String representing the name of the database schema.
<i>dbTable</i>	String representing the base name of the database table. The name of the geometry for that table will be prepended to the base name.

**See also:**

```
    sosicon::ConverterSosi2psql::insertPoint()
    sosicon::ConverterSosi2psql::insertLineString()
    sosicon::ConverterSosi2psql::insertPolygon()
```

Definition at line 517 of file converter\_sosi2psql.cpp.

```
bool sosicon::ConverterSosi2psql::objTypeExcluded (sosi::SosiElementSearch &
src)[private]
```

Test if current element is filtered out by -t parameter.

If the user uses the -t parameter to specify which OBJTYPE elements to include in the export, this function tests if current element is opted out of the export.

**Parameters:**

<i>src</i>	SOSI element serch result to test.
------------	------------------------------------

**Returns:**

True if current element should be excluded from the export file.

Definition at line 553 of file converter\_sosi2psql.cpp.

```
void sosicon::ConverterSosi2psql::run (bool *   cancel = 0x00)[virtual]
```

Start conversion.

Implementation details in **sosicon::IConverter::run()**

**See also:**

```
    sosicon::IConverter::run()
```

Implements **sosicon::IConverter** (p.83).

Definition at line 560 of file converter\_sosi2psql.cpp.

```
void sosicon::ConverterSosi2psql::writePsql (std::string  sridDest, std::string  dbSchema,  
std::string  dbTable)[private]
```

Write SQL content.

Assembles and prepares the SQL statements before writing them to the destination file.

**Parameters:**

<i>sridDest</i>	Spatial reference grid ID for the target file.
<i>dbSchema</i>	String representing the name of the database schema.
<i>dbTable</i>	String representing the base name of the database table. The name of the geometry for that table will be prepended to the base name.

Definition at line 614 of file converter\_sosi2psql.cpp.

## Member Data Documentation

**CommandLine\* sosicon::ConverterSosi2psql::mCmd [private]**

Command line wrapper.

Definition at line 98 of file converter\_sosi2psql.h.

**std::string sosicon::ConverterSosi2psql::mCurrentSourcefile [private]**

Source file currently in process.

Definition at line 101 of file converter\_sosi2psql.h.

**FieldsListCollection sosicon::ConverterSosi2psql::mFieldsListCollection [private]**

Collection of fields, one item for each geometry type.

Definition at line 104 of file converter\_sosi2psql.h.

**RowsListCollection sosicon::ConverterSosi2psql::mRowsListCollection [private]**

Collection of rows, one item for each geometry type.

Definition at line 107 of file converter\_sosi2psql.h.

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi2psql.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi2psql.cpp**

## sosicon::ConverterSosi2shp Class Reference

SOSI to ESRI Shape converter.

```
#include <converter_sosi2shp.h>
```

Inheritance diagram for sosicon::ConverterSosi2shp:



### Public Member Functions

- **ConverterSosi2shp ()**  
*Constructor.*
- virtual **~ConverterSosi2shp ()**  
*Destructor.*
- virtual void **init (CommandLine \*cmd)**  
*Initialize converter.*
- virtual void **run** (bool \*cancel=0x00)  
*Start conversion.*

### Private Member Functions

- template<typename T > void **writeFile** (shape::Shapefile &shp, std::string basePath, std::string extension)  
*Save specific shapefile part.*
- void **makeShp** (ISosiElement \*sosiTree, bool \*cancel)
- std::string **makeBasePath** (std::string objTypeName)  
*Make base file path for destination files.*

### Private Attributes

- **CommandLine \* mCmd**  
*Command line wrapper.*
- std::string **mCurrentSourcefile**  
*Souce file currently in process.*

---

## Detailed Description

SOSI to ESRI Shape converter.

If command-line parameter -2shp is specified, this converter will handle the output generation. Produces an ESRI Shape-file from SOSI source.

Definition at line 60 of file converter\_sosi2shp.h.

---

## Constructor & Destructor Documentation

**sosicon::ConverterSosi2shp::ConverterSosi2shp ()** `[inline]`

Constructor.

Definition at line 117 of file converter\_sosi2shp.h.

**virtual sosicon::ConverterSosi2shp::~~ConverterSosi2shp ()** `[inline], [virtual]`

Destructor.

Definition at line 120 of file converter\_sosi2shp.h.

---

## Member Function Documentation

**virtual void sosicon::ConverterSosi2shp::init (CommandLine \* *cmd*)** `[inline], [virtual]`

Initialize converter.

Implementation details in **sosicon::IConverter::init()**

### See also:

**sosicon::IConverter::init()**

Implements **sosicon::IConverter** (p.82).

Definition at line 127 of file converter\_sosi2shp.h.

**std::string sosicon::ConverterSosi2shp::makeBasePath (std::string *objTypeName*)** `[private]`

Make base file path for destination files.

If the user specified an output file name, it will be used as a candidate for a base name to create shp, shx and dbf files for the shape export. Otherwise, the name of the first source file will be used by default.

This function checks if there are any name collisions, incrementing a postfixed number to the base name until a unique name is found.

### Returns:

Modified, unique destination base name with directory (if provided), without file name extension.

Definition at line 126 of file converter\_sosi2shp.cpp.

**void sosicon::ConverterSosi2shp::makeShp (ISosiElement \* *sosiTree*, bool \* *cancel*)** `[private]`

Definition at line 21 of file converter\_sosi2shp.cpp.

**void sosicon::ConverterSosi2shp::run (bool \* *cancel* = 0x00)** `[virtual]`

Start conversion.



Implementation details in **sosicon::IConverter::run()**

**See also:**

**sosicon::IConverter::run()**

Implements **sosicon::IConverter** (p.83).

Definition at line 174 of file converter\_sosi2shp.cpp.

**template<typename T > void sosicon::ConverterSosi2shp::writeFile (shape::Shapefile & shp, std::string basePath, std::string extension)[inline], [private]**

Save specific shapefile part.

The shapefile format consists of several files. Use corresponding interface to cast a instance of **IShapefile** to the correct file part for writing. The shapefile parts interfaces are:

- **IShapefileShpPart**
- **IShapefileShxPart**
- **IShapefileDbfPart**
- **IShapefilePrjPart**

**Parameters:**

<i>shp</i>	Reference to the source ShapeFile instance.
<i>basePath</i>	Path and file title for the file to be written, without extension.
<i>extension</i>	additional file extensions to be appended before the main extension, which is one of the following: <ul style="list-style-type: none"><li>• shp (shapefile part)</li><li>• shx (index part)</li><li>• dbf (attributes part)</li><li>• prj (projection part)</li></ul>

Definition at line 83 of file converter\_sosi2shp.h.

---

## Member Data Documentation

**CommandLine\* sosicon::ConverterSosi2shp::mCmd [private]**

Command line wrapper.

Definition at line 93 of file converter\_sosi2shp.h.

**std::string sosicon::ConverterSosi2shp::mCurrentSourcefile [private]**

Souce file currently in process.

Definition at line 96 of file converter\_sosi2shp.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi2shp.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi2shp.cpp**



## sosicon::ConverterSosi2tsv Class Reference

SOSI to TSV converter.

```
#include <converter_sosi2tsv.h>
```

Inheritance diagram for sosicon::ConverterSosi2tsv:



### Public Member Functions

- **ConverterSosi2tsv ()**  
*Constructor.*
- virtual void **init** (**CommandLine** \*cmd)  
*Initialize converter.*
- virtual void **run** (bool \*cancel=0x00)  
*Start conversion.*

### Private Member Functions

- virtual ~**ConverterSosi2tsv** ()  
*Destructor.*

### Private Attributes

- **CommandLine** \* mCmd  
*Command line wrapper.*

---

## Detailed Description

SOSI to TSV converter.

If command-line parameter -2tsv is specified, this converter will handle the output generation. Produces a TSV file (tab separated values) SOSI source.

Definition at line 39 of file `converter_sosi2tsv.h`.

---

## Constructor & Destructor Documentation

**virtual sosicon::ConverterSosi2tsv::~~ConverterSosi2tsv ()** [inline], [private], [virtual]

Destructor.

Definition at line 45 of file `converter_sosi2tsv.h`.

**sosicon::ConverterSosi2tsv::ConverterSosi2tsv ()** [inline]

Constructor.

Definition at line 49 of file converter\_sosi2tsv.h.

---

## Member Function Documentation

**virtual void sosicon::ConverterSosi2tsv::init (CommandLine \* *cmd*)***[inline]*, *[virtual]*

Initialize converter.

Implementation details in **sosicon::IConverter::init()**

**See also:**

**sosicon::IConverter::init()**

Implements **sosicon::IConverter** (p.82).

Definition at line 56 of file converter\_sosi2tsv.h.

**void sosicon::ConverterSosi2tsv::run (bool \* *cancel* = 0x00)***[virtual]*

Start conversion.

Implementation details in **sosicon::IConverter::run()**

**See also:**

**sosicon::IConverter::run()**

Implements **sosicon::IConverter** (p.83).

Definition at line 21 of file converter\_sosi2tsv.cpp.

---

## Member Data Documentation

**CommandLine\* sosicon::ConverterSosi2tsv::mCmd***[private]*

Command line wrapper.

Definition at line 42 of file converter\_sosi2tsv.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi2tsv.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi2tsv.cpp**

## sosicon::ConverterSosi2xml Class Reference

SOSI to ESRI Shape converter.

```
#include <converter_sosi2xml.h>
```

Inheritance diagram for sosicon::ConverterSosi2xml:



### Public Member Functions

- **ConverterSosi2xml ()**  
*Constructor.*
- virtual **~ConverterSosi2xml ()**  
*Destructor.*
- virtual void **init** (**CommandLine** \*cmd)  
*Initialize converter.*
- virtual void **run** (bool \*cancel=0x00)  
*Start conversion.*

### Private Member Functions

- void **makeXML** (**ISosiElement** \*parent)

### Private Attributes

- **CommandLine** \* mCmd  
*Command line wrapper.*

---

## Detailed Description

SOSI to ESRI Shape converter.

If command-line parameter -2xml is specified, this converter will handle the output generation. Produces an ESRI Shape-file from SOSI source.

Definition at line 41 of file `converter_sosi2xml.h`.

---

## Constructor & Destructor Documentation

**sosicon::ConverterSosi2xml::ConverterSosi2xml ()** `[inline]`

Constructor.

Definition at line 51 of file `converter_sosi2xml.h`.

**virtual sosicon::ConverterSosi2xml::~~ConverterSosi2xml () [inline], [virtual]**

Destructor.

Definition at line 54 of file converter\_sosi2xml.h.

---

## Member Function Documentation

**virtual void sosicon::ConverterSosi2xml::init (CommandLine \* *cmd*) [inline], [virtual]**

Initialize converter.

Implementation details in **sosicon::IConverter::init()**

**See also:**

**sosicon::IConverter::init()**

Implements **sosicon::IConverter** (p.82).

Definition at line 61 of file converter\_sosi2xml.h.

**void sosicon::ConverterSosi2xml::makeXML (ISosiElement \* *parent*) [private]**

Definition at line 21 of file converter\_sosi2xml.cpp.

**void sosicon::ConverterSosi2xml::run (bool \* *cancel* = 0x00) [virtual]**

Start conversion.

Implementation details in **sosicon::IConverter::run()**

**See also:**

**sosicon::IConverter::run()**

Implements **sosicon::IConverter** (p.83).

Definition at line 26 of file converter\_sosi2xml.cpp.

---

## Member Data Documentation

**CommandLine\* sosicon::ConverterSosi2xml::mCmd [private]**

Command line wrapper.

Definition at line 44 of file converter\_sosi2xml.h.

---

**The documentation for this class was generated from the following files:**

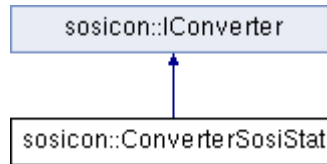
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi2xml.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi2xml.cpp**

## sosicon::ConverterSosiStat Class Reference

SOSI to ESRI Shape converter.

```
#include <converter_sosi_stat.h>
```

Inheritance diagram for sosicon::ConverterSosiStat:



### Public Member Functions

- **ConverterSosiStat** ()  
*Constructor.*
- virtual **~ConverterSosiStat** ()  
*Destructor.*
- virtual void **init** (**CommandLine** \*cmd)  
*Initialize converter.*
- virtual void **run** (bool \*cancel=0x00)  
*Start conversion.*

### Private Member Functions

- void **printElementData** (**ISosiElement** \*e, **sosi::SosiElementSearch** src, int padding)  
*Output simple element attributes.*
- void **printListContent** (std::map< std::string, int > list, int padding)  
*Output content of map<string,int>*
- void **printTableHeader** (std::string col1, std::string col2, int padding)  
*Output simple element attributes.*
- void **makeStat** (**ISosiElement** \*parent)  
*Output table header with column titles.*

### Private Attributes

- **CommandLine** \* **mCmd**  
*Command line wrapper.*
- std::map< std::string, int > **mObjTypes**  
*Map keeping count of objtypes.*
- std::map< std::string, int > **mGeoTypes**  
*Map keeping count of geometry.*

---

### Detailed Description

SOSI to ESRI Shape converter.

If command-line parameter -stat is specified, this converter will handle the output generation. Produces an ESRI Shape-file from SOSI source.

Definition at line 52 of file converter\_sosi\_stat.h.

---

## Constructor & Destructor Documentation

**sosicon::ConverterSosiStat::ConverterSosiStat ()** `[inline]`

Constructor.

Definition at line 78 of file converter\_sosi\_stat.h.

**virtual sosicon::ConverterSosiStat::~~ConverterSosiStat ()** `[inline], [virtual]`

Destructor.

Definition at line 81 of file converter\_sosi\_stat.h.

---

## Member Function Documentation

**virtual void sosicon::ConverterSosiStat::init (CommandLine \* *cmd*)** `[inline], [virtual]`

Initialize converter.

Implementation details in **sosicon::IConverter::init()**

**See also:**

**sosicon::IConverter::init()**

Implements **sosicon::IConverter** (p.82).

Definition at line 88 of file converter\_sosi\_stat.h.

**void sosicon::ConverterSosiStat::makeStat (ISosiElement \* *parent*)** `[private]`

Output table header with column titles.

Definition at line 21 of file converter\_sosi\_stat.cpp.

**void sosicon::ConverterSosiStat::printElementData (ISosiElement \* *e*, sosi::SosiElementSearch *src*, int *padding*)** `[private]`

Output simple element attributes.

Definition at line 46 of file converter\_sosi\_stat.cpp.

**void sosicon::ConverterSosiStat::printListContent (std::map< std::string, int > *list*, int *padding*)** `[private]`

Output content of map<string,int>

Definition at line 58 of file converter\_sosi\_stat.cpp.



**void sosicon::ConverterSosiStat::printTableHeader (std::string col1, std::string col2, int padding) [private]**

Output simple element attributes.

Definition at line 70 of file converter\_sosi\_stat.cpp.

**void sosicon::ConverterSosiStat::run (bool \* cancel = 0x00) [virtual]**

Start conversion.

Implementation details in **sosicon::IConverter::run()**

**See also:**

**sosicon::IConverter::run()**

Implements **sosicon::IConverter** (p.83).

Definition at line 77 of file converter\_sosi\_stat.cpp.

---

## Member Data Documentation

**CommandLine\* sosicon::ConverterSosiStat::mCmd [private]**

Command line wrapper.

Definition at line 55 of file converter\_sosi\_stat.h.

**std::map<std::string, int> sosicon::ConverterSosiStat::mGeoTypes [private]**

Map keeping count of geometry.

Definition at line 61 of file converter\_sosi\_stat.h.

**std::map<std::string, int> sosicon::ConverterSosiStat::mObjTypes [private]**

Map keeping count of objtypes.

Definition at line 58 of file converter\_sosi\_stat.h.

---

**The documentation for this class was generated from the following files:**

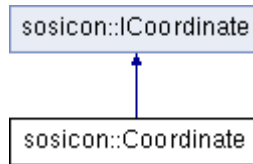
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi\_stat.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi\_stat.cpp**

## sosicon::Coordinate Class Reference

**Coordinate** container.

```
#include <coordinate.h>
```

Inheritance diagram for sosicon::Coordinate:



### Public Member Functions

- virtual `~Coordinate ()`
- **Coordinate ()**
- virtual double **getE ()**  
*Get east coordinate.*
- virtual double **getN ()**  
*Get north coordinate.*
- virtual bool **leftOf (ICoordinate \*c)**  
*Test if this coordinate is to the left of another.*
- virtual bool **rightOf (ICoordinate \*c)**  
*Test if this coordinate is to the right of another.*
- virtual void **setE** (double coordEast)  
*Set east coordinate.*
- virtual void **setN** (double coordNorth)  
*Set north coordinate.*
- virtual void **setH** (double altitude)  
*Set altitude.*
- virtual void **shift** (int offsetN, int offsetE)  
*Shift coordinate by specified offset.*
- virtual void **divide** (int divisor)  
*Divide coordinate by specified divisor.*
- virtual bool **equals (ICoordinate \*c)**  
*Check if two points match.*
- virtual std::string **toString ()**  
*Make string representation.*

### Private Attributes

- double **mEast**
- double **mNorth**
- double **mAltitude**

---

## Detailed Description

**Coordinate** container.

**Author:**

Espen Andersen

**Copyright:**

GNU General Public License

Stores a geographical position with some additional information.

Definition at line 36 of file coordinate.h.

---

## Constructor & Destructor Documentation

**virtual sosicon::Coordinate::~~Coordinate () [inline], [virtual]**

Definition at line 44 of file coordinate.h.

**sosicon::Coordinate::Coordinate () [inline]**

Definition at line 45 of file coordinate.h.

---

## Member Function Documentation

**virtual void sosicon::Coordinate::divide (int *divisor*) [inline], [virtual]**

Divide coordinate by specified divisor.

Implements **sosicon::ICoordinate** (p.85).

Definition at line 54 of file coordinate.h.

**virtual bool sosicon::Coordinate::equals (ICoordinate \* *c*) [inline], [virtual]**

Check if two points match.

Implements **sosicon::ICoordinate** (p.85).

Definition at line 55 of file coordinate.h.

**virtual double sosicon::Coordinate::getE () [inline], [virtual]**

Get east coordinate.

Implements **sosicon::ICoordinate** (p.85).

Definition at line 46 of file coordinate.h.

**virtual double sosicon::Coordinate::getN () [inline], [virtual]**

Get north coordinate.

Implements **sosicon::ICoordinate** (p.85).

Definition at line 47 of file coordinate.h.

**virtual bool sosicon::Coordinate::leftOf (ICoordinate \* c)[inline], [virtual]**

Test if this coordinate is to the left of another.

Implements **sosicon::ICoordinate** (p.85).

Definition at line 48 of file coordinate.h.

**virtual bool sosicon::Coordinate::rightOf (ICoordinate \* c)[inline], [virtual]**

Test if this coordinate is to the right of another.

Implements **sosicon::ICoordinate** (p.85).

Definition at line 49 of file coordinate.h.

**virtual void sosicon::Coordinate::setE (double coordEast)[inline], [virtual]**

Set east coordinate.

Implements **sosicon::ICoordinate** (p.86).

Definition at line 50 of file coordinate.h.

**virtual void sosicon::Coordinate::setH (double altitude)[inline], [virtual]**

Set altitude.

Implements **sosicon::ICoordinate** (p.86).

Definition at line 52 of file coordinate.h.

**virtual void sosicon::Coordinate::setN (double coordNorth)[inline], [virtual]**

Set north coordinate.

Implements **sosicon::ICoordinate** (p.86).

Definition at line 51 of file coordinate.h.

**virtual void sosicon::Coordinate::shift (int offsetN, int offsetE)[inline], [virtual]**

Shift coordinate by specified offset.

Implements **sosicon::ICoordinate** (p.86).

Definition at line 53 of file coordinate.h.

**virtual std::string sosicon::Coordinate::toString () [inline], [virtual]**

Make string representation.

Implements **sosicon::ICoordinate** (p.86).

Definition at line 56 of file coordinate.h.

---

## Member Data Documentation

**double sosicon::Coordinate::mAltitude [private]**

Definition at line 40 of file coordinate.h.

**double sosicon::Coordinate::mEast [private]**

Definition at line 38 of file coordinate.h.

**double sosicon::Coordinate::mNorth [private]**

Definition at line 39 of file coordinate.h.

---

**The documentation for this class was generated from the following file:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**coordinate.h**

## sosicon::CoordinateCollection Class Reference

**Coordinate** container.

```
#include <coordinate_collection.h>
```

### Public Member Functions

- virtual **~CoordinateCollection** ()  
*Destructor.*
- **CoordinateCollection** ()  
*Constructor.*
- void **free** ()  
*Free allocated memory.*
- void **discoverCoords** (ISosiElement \*sosi)  
*Extracts coordinates from SOSI element.*
- bool **getNextInGeom** (ICoordinate \*&coord)  
*Retrieve next coordinate in collection.*
- std::vector< ICoordinate \* > & **getGeom** ()
- std::vector< int > & **getGeomSizes** ()
- int **getNumPointsGeom** ()
- int **getNumPartsGeom** ()
- std::vector< ICoordinate \* > & **getHoles** ()
- std::vector< int > & **getHoleSizes** ()
- int **getNumPointsHoles** ()
- int **getNumPartsHoles** ()
- double **getXmin** ()
- double **getYmin** ()
- double **getXmax** ()
- double **getYmax** ()

### Private Member Functions

- void **extractPath** (ISosiElement \*referencedElement, bool reverse, int &numPoints, sosi::NorthEastList &target)  
*Get coordinate values from SOSI element.*

### Private Attributes

- sosi::NorthEastList **mGeom**  
*Stores collection of pointers to coordinates for geometries.*
- std::vector< ICoordinate \* > **mGeomNormalized**
- sosi::NorthEastList **mHoles**
- std::vector< ICoordinate \* > **mHolesNormalized**
- int **mNumPartsGeom**
- int **mNumPartsHoles**
- int **mNumPointsGeom**
- int **mNumPointsHoles**
- std::vector< int > **mGeomSizes**
- std::vector< int > **mHoleSizes**
- sosi::NorthEastList::iterator **mGeomIndex**
- double **mXmin**

- double **mYmin**
- double **mXmax**
- double **mYmax**

---

## Detailed Description

**Coordinate** container.

### Author:

Espen Andersen

### Copyright:

GNU General Public License

Stores a collection of geographical positions.

Definition at line 77 of file `coordinate_collection.h`.

---

## Constructor & Destructor Documentation

**sosicon::CoordinateCollection::~~CoordinateCollection () [virtual]**

Destructor.

Definition at line 84 of file `coordinate_collection.cpp`.

**sosicon::CoordinateCollection::CoordinateCollection () [inline]**

Constructor.

Definition at line 114 of file `coordinate_collection.h`.

---

## Member Function Documentation

**void sosicon::CoordinateCollection::discoverCoords (ISosiElement \* *sosi*)**

Extracts coordinates from SOSI element.

This method retrieves the physical coordinates for a SOSI geometry, if applicable, and populates the coordinate collection.

This algorithm resolves referenced objects for polygons and presents the coordinates in correct order.

### Parameters:

<i>sosi</i>	SOSI element from which to extract coordinates.
-------------	---

Definition at line 95 of file `coordinate_collection.cpp`.

**void sosicon::CoordinateCollection::extractPath (ISosiElement \* *referencedElement*, bool *reverse*, int & *numPoints*, sosicon::NorthEastList & *target*) [private]**

Get coordinate values from SOSI element.

Definition at line 161 of file coordinate\_collection.cpp.

**void sosicon::CoordinateCollection::free ()**

Free allocated memory.

Definition at line 89 of file coordinate\_collection.cpp.

**std::vector< sosicon::ICoordinate \* > & sosicon::CoordinateCollection::getGeom ()**

Definition at line 191 of file coordinate\_collection.cpp.

**std::vector<int>& sosicon::CoordinateCollection::getGeomSizes () [inline]**

Definition at line 151 of file coordinate\_collection.h.

**std::vector< sosicon::ICoordinate \* > & sosicon::CoordinateCollection::getHoles ()**

Definition at line 205 of file coordinate\_collection.cpp.

**std::vector<int>& sosicon::CoordinateCollection::getHoleSizes () [inline]**

Definition at line 156 of file coordinate\_collection.h.

**bool sosicon::CoordinateCollection::getNextInGeom (ICoordinate \*& coord)**

Retrieve next coordinate in collection.

Iterates through the coordinate list until it reaches the end, passing a pointer to the next element to the coord reference. The value of coord must be zero on the first pass in order to start the iteration on the first **ISosiElement**.

**Returns:**

The function returns true if there are more coordinates in the collection, or false if the last coordinate is encountered.

Definition at line 228 of file coordinate\_collection.cpp.

**int sosicon::CoordinateCollection::getNumPartsGeom () [inline]**

Definition at line 153 of file coordinate\_collection.h.

**int sosicon::CoordinateCollection::getNumPartsHoles () [inline]**

Definition at line 158 of file coordinate\_collection.h.



**int sosicon::CoordinateCollection::getNumPointsGeom () [inline]**

Definition at line 152 of file coordinate\_collection.h.

**int sosicon::CoordinateCollection::getNumPointsHoles () [inline]**

Definition at line 157 of file coordinate\_collection.h.

**double sosicon::CoordinateCollection::getXmax () [inline]**

Definition at line 164 of file coordinate\_collection.h.

**double sosicon::CoordinateCollection::getXmin () [inline]**

Definition at line 160 of file coordinate\_collection.h.

**double sosicon::CoordinateCollection::getYmax () [inline]**

Definition at line 166 of file coordinate\_collection.h.

**double sosicon::CoordinateCollection::getYmin () [inline]**

Definition at line 162 of file coordinate\_collection.h.

---

## Member Data Documentation

**sosi::NorthEastList sosicon::CoordinateCollection::mGeom [private]**

Stores collection of pointers to coordinates for geometries.

Definition at line 80 of file coordinate\_collection.h.

**sosi::NorthEastList::iterator sosicon::CoordinateCollection::mGeomIndex [private]**

Definition at line 95 of file coordinate\_collection.h.

**std::vector<ICoordinate\*> sosicon::CoordinateCollection::mGeomNormalized [private]**

Definition at line 81 of file coordinate\_collection.h.

**std::vector<int> sosicon::CoordinateCollection::mGeomSizes [private]**

Definition at line 92 of file coordinate\_collection.h.

**sosi::NorthEastList sosicon::CoordinateCollection::mHoles [private]**

Definition at line 83 of file coordinate\_collection.h.

**std::vector<int> sosicon::CoordinateCollection::mHoleSizes [private]**

Definition at line 93 of file coordinate\_collection.h.

**std::vector<ICoordinate\*> sosicon::CoordinateCollection::mHolesNormalized [private]**

Definition at line 84 of file coordinate\_collection.h.

**int sosicon::CoordinateCollection::mNumPartsGeom [private]**

Definition at line 86 of file coordinate\_collection.h.

**int sosicon::CoordinateCollection::mNumPartsHoles [private]**

Definition at line 87 of file coordinate\_collection.h.

**int sosicon::CoordinateCollection::mNumPointsGeom [private]**

Definition at line 89 of file coordinate\_collection.h.

**int sosicon::CoordinateCollection::mNumPointsHoles [private]**

Definition at line 90 of file coordinate\_collection.h.

**double sosicon::CoordinateCollection::mXmax [private]**

Definition at line 99 of file coordinate\_collection.h.

**double sosicon::CoordinateCollection::mXmin [private]**

Definition at line 97 of file coordinate\_collection.h.

**double sosicon::CoordinateCollection::mYmax [private]**

Definition at line 100 of file coordinate\_collection.h.

**double sosicon::CoordinateCollection::mYmin [private]**

Definition at line 98 of file coordinate\_collection.h.

---

**The documentation for this class was generated from the following files:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate_collection.h`
- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate_collection.cpp`

## sosicon::sosi::CoordSys Class Reference

SOSI coordinate system.

```
#include <sosi_types.h>
```

### Public Member Functions

- **CoordSys** ()
- **CoordSys** (int sysCode, std::string **srid**, std::string **displayString**, std::string **prjString**)
- std::string **displayString** ()
- std::string **prjString** ()
- std::string **srid** ()
- bool **valid** ()

### Private Attributes

- int **mSysCode**  
*SOSI SYSKODE.*
- std::string **mSrid**  
*EPSG SRID.*
- std::string **mPrjString**  
*Projection string.*
- std::string **mDisplayString**  
*Display string.*

---

### Detailed Description

SOSI coordinate system.

Definition at line 152 of file `sosi_types.h`.

---

### Constructor & Destructor Documentation

**sosicon::sosi::CoordSys::CoordSys** () [inline]

Definition at line 161 of file `sosi_types.h`.

**sosicon::sosi::CoordSys::CoordSys** (int sysCode, std::string srid, std::string displayString, std::string prjString) [inline]

Definition at line 163 of file `sosi_types.h`.

---

### Member Function Documentation

std::string **sosicon::sosi::CoordSys::displayString** () [inline]

Definition at line 171 of file sosi\_types.h.

**std::string sosicon::sosi::CoordSys::prjString () [inline]**

Definition at line 173 of file sosi\_types.h.

**std::string sosicon::sosi::CoordSys::srid () [inline]**

Definition at line 175 of file sosi\_types.h.

**bool sosicon::sosi::CoordSys::valid () [inline]**

Definition at line 177 of file sosi\_types.h.

---

## Member Data Documentation

**std::string sosicon::sosi::CoordSys::mDisplayString [private]**

Display string.

Definition at line 157 of file sosi\_types.h.

**std::string sosicon::sosi::CoordSys::mPrjString [private]**

Projection string.

Definition at line 156 of file sosi\_types.h.

**std::string sosicon::sosi::CoordSys::mSrid [private]**

EPSG SRID.

Definition at line 155 of file sosi\_types.h.

**int sosicon::sosi::CoordSys::mSysCode [private]**

SOSI SYSKODE.

Definition at line 154 of file sosi\_types.h.

---

**The documentation for this class was generated from the following file:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_types.h

## sosicon::shape::DoubleField Union Reference

32 bit double / byte field

```
#include <shapefile_types.h>
```

### Public Attributes

- double **d**
  - char **b** [sizeof(double)]
- 

### Detailed Description

32 bit double / byte field

Definition at line 78 of file shapefile\_types.h.

---

### Member Data Documentation

**char sosicon::shape::DoubleField::b[sizeof(double)]**

Definition at line 80 of file shapefile\_types.h.

**double sosicon::shape::DoubleField::d**

Definition at line 79 of file shapefile\_types.h.

---

The documentation for this union was generated from the following file:

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile\_types.h

## sosicon::EventDispatcher< Event > Class Template Reference

Event dispatcher template class.

```
#include <event_dispatcher.h>
```

### Classes

- class **Listener**

### Public Types

- typedef std::vector< **Listener** \* > **ListenerLst**

### Public Member Functions

- void **addEventListener** (**Listener** \*listener)
- void **removeEventListener** (**Listener** \*listener)
- void **Dispatch** (Event &e)

### Private Attributes

- **ListenerLst** mListeners

---

## Detailed Description

template<typename Event>

class sosicon::EventDispatcher< Event >

Event dispatcher template class.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Implements event dispatcher for specified event type.

Definition at line 35 of file event\_dispatcher.h.

---

## Member Typedef Documentation

template<typename Event> typedef std::vector<Listener\*> sosicon::EventDispatcher< Event >::ListenerLst

Definition at line 47 of file event\_dispatcher.h.

## Member Function Documentation

**template<typename Event> void sosicon::EventDispatcher< Event >::addEventListener (Listener \* *listener*) [inline]**

Definition at line 55 of file event\_dispatcher.h.

**template<typename Event> void sosicon::EventDispatcher< Event >::Dispatch (Event & *e*) [inline]**

Definition at line 68 of file event\_dispatcher.h.

**template<typename Event> void sosicon::EventDispatcher< Event >::removeEventListener (Listener \* *listener*) [inline]**

Definition at line 61 of file event\_dispatcher.h.

---

## Member Data Documentation

**template<typename Event> ListenerLst sosicon::EventDispatcher< Event >::mListeners [private]**

Definition at line 51 of file event\_dispatcher.h.

---

**The documentation for this class was generated from the following file:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/event\_dispatcher.h



## sosicon::Factory Class Reference

**Factory** class.

```
#include <factory.h>
```

### Static Public Member Functions

- static void **get** (**IConverter** \*&converter, **CommandLine** \*cmd)  
*Retrieve converter.*
- static void **release** (**IConverter** \*&converter)  
*Releases converter.*

---

### Detailed Description

**Factory** class.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Responsible for creating an appropriate **IConverter** implementation instance. Uses the command-line arguments to decide what type of converter is required.

Definition at line 38 of file factory.h.

---

### Member Function Documentation

**void sosicon::Factory::get** (sosicon::IConverter \*& *converter*, sosicon::CommandLine \**cmd*)**[static]**

Retrieve converter.

Selects an **IConverter** implementation and creates an object of that class based on the user's command-line arguments.

#### Note:

Any object delivered with **Factory::get()** must be freed by calling **Factory::release()**

#### See also:

**Factory::release()**

#### Parameters:

<i>converter</i>	Reference to the pointer to receive the new <b>IConverter</b> .
<i>cmd</i>	<b>CommandLine</b> object with current command-line arguments.

Definition at line 21 of file factory.cpp.

**void sosicon::Factory::release** (sosicon::IConverter \*& *converter*)**[static]**

Releases converter.

Frees allocated resources and releases **IConverter** object. Any object retrieved from **Factory::get()** must be disposed of through this function.

**Parameters:**

<i>converter</i>	Reference to a pointer containing the <b>IConverter</b> instance to be released. The pointer will be reset to 0 after object deletion.
------------------	--

Definition at line 45 of file factory.cpp.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**factory.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**factory.cpp**

## **sosicon::ConverterSosi2psql::Field Class Reference**

### **Public Member Functions**

- `bool isNumeric ()`
- `std::string::size_type length ()`
- `Field ()`
- `Field (std::string &str)`
- `std::string::size_type expand (std::string &str)`

### **Private Attributes**

- `std::string::size_type mMaxLength`
- `std::string::size_type mMinLength`
- `bool mIsNumeric`

---

### **Detailed Description**

Definition at line 54 of file `converter_sosi2psql.h`.

---

### **Constructor & Destructor Documentation**

**sosicon::ConverterSosi2psql::Field::Field () [inline]**

Definition at line 71 of file `converter_sosi2psql.h`.

**sosicon::ConverterSosi2psql::Field::Field (std::string & str) [inline]**

Definition at line 75 of file `converter_sosi2psql.h`.

---

### **Member Function Documentation**

**std::string::size\_type sosicon::ConverterSosi2psql::Field::expand (std::string & str) [inline]**

Definition at line 81 of file `converter_sosi2psql.h`.

**bool sosicon::ConverterSosi2psql::Field::isNumeric () [inline]**

Definition at line 59 of file `converter_sosi2psql.h`.

**std::string::size\_type sosicon::ConverterSosi2psql::Field::length () [inline]**

Definition at line 68 of file `converter_sosi2psql.h`.

---

## Member Data Documentation

**bool sosicon::ConverterSosi2psql::Field::mIsNumeric** [private]

Definition at line 57 of file converter\_sosi2psql.h.

**std::string::size\_type sosicon::ConverterSosi2psql::Field::mMaxLength** [private]

Definition at line 55 of file converter\_sosi2psql.h.

**std::string::size\_type sosicon::ConverterSosi2psql::Field::mMinLength** [private]

Definition at line 56 of file converter\_sosi2psql.h.

---

**The documentation for this class was generated from the following file:**

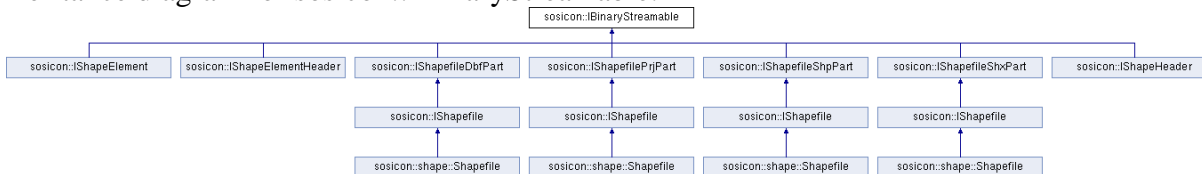
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**converter\_sosi2psql.h**

## sosicon::IBinaryStreamable Class Reference

Interface: Binary streamable object.

```
#include <i_binary_streamable.h>
```

Inheritance diagram for sosicon::IBinaryStreamable:



### Public Member Functions

- virtual **~IBinaryStreamable** ()  
*Destructor.*
- virtual void **writeBinary** (std::ostream &os)=0  
*Writes binary data to output stream.*

---

### Detailed Description

Interface: Binary streamable object.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 34 of file i\_binary\_streamable.h.

---

### Constructor & Destructor Documentation

**virtual sosicon::IBinaryStreamable::~~IBinaryStreamable () [inline], [virtual]**

Destructor.

Definition at line 38 of file i\_binary\_streamable.h.

---

### Member Function Documentation

**virtual void sosicon::IBinaryStreamable::writeBinary (std::ostream & os) [pure virtual]**

Writes binary data to output stream.

Invoked when the operator<< is used to stream to an std::ostream object. Binary write operation is performed for current implementation here.

**Parameters:**

<i>os</i>	Target stream object.
-----------	-----------------------

Implemented in **sosicon::IShapefileDbfPart** (p.103), **sosicon::IShapefilePrjPart** (p.105), **sosicon::IShapefileShpPart** (p.107), and **sosicon::IShapefileShxPart** (p.109).

---

**The documentation for this class was generated from the following file:**

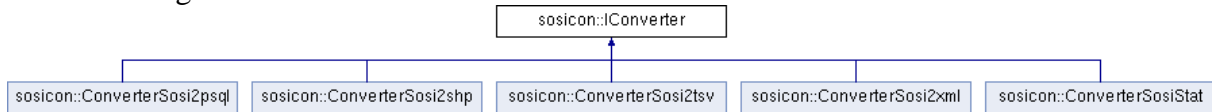
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_binary\_streamable.h

## sosicon::IConverter Class Reference

Interface: Converter.

```
#include <i_converter.h>
```

Inheritance diagram for sosicon::IConverter:



### Public Member Functions

- virtual **~IConverter** ()  
*Destructor.*
- virtual void **init** (**CommandLine** \*cmd)=0  
*Initialize converter.*
- virtual void **run** (bool \*cancel=0x00)=0  
*Start conversion.*

---

### Detailed Description

Interface: Converter.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Represents the generic form of a converter. The factory class is responsible for creating a converter based upon input parameters. The returned object is then interacted on through this interface.

Definition at line 38 of file i\_converter.h.

---

### Constructor & Destructor Documentation

**virtual sosicon::IConverter::~~IConverter () [inline], [virtual]**

Destructor.

Definition at line 43 of file i\_converter.h.

---

### Member Function Documentation

**virtual void sosicon::IConverter::init (CommandLine \* cmd)[pure virtual]**

Initialize converter.

Conversion setup. Uses the **CommandLine** to determine what operations to perform.

**Parameters:**

<i>cmd</i>	Arguments from the command-line parsed and ready within a <b>sosicon::CommandLine</b> object.
------------	---

Implemented in **sosicon::ConverterSosi2psql** (p.46), **sosicon::ConverterSosi2shp** (p.51), **sosicon::ConverterSosiStat** (p.59), **sosicon::ConverterSosi2xml** (p.57), and **sosicon::ConverterSosi2tsv** (p.55).

**virtual void sosicon::IConverter::run (bool \* *cancel* = 0x00)[pure virtual]**

Start conversion.

Run the conversion routine. Outputs the destination file according to the preferences given from the command-line.

**Parameters:**

<i>cancel</i>	If sosicon runs in a worker thread, this parameter governs whether the conversion process should be aborted prematurely.
---------------	--

Implemented in **sosicon::ConverterSosi2psql** (p.48), **sosicon::ConverterSosi2shp** (p.51), **sosicon::ConverterSosiStat** (p.60), **sosicon::ConverterSosi2xml** (p.57), and **sosicon::ConverterSosi2tsv** (p.55).

---

**The documentation for this class was generated from the following file:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_converter.h

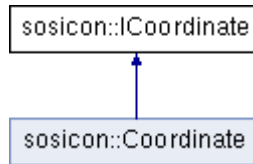


## sosicon::ICoordinate Class Reference

Interface: **Coordinate**.

```
#include <i_coordinate.h>
```

Inheritance diagram for sosicon::ICoordinate:



### Public Member Functions

- virtual **~ICoordinate** ()  
*Destructor.*
- virtual double **getE** ()=0  
*Get east coordinate.*
- virtual double **getN** ()=0  
*Get north coordinate.*
- virtual bool **leftOf** (**ICoordinate** \*c)=0  
*Test if this coordinate is to the left of another.*
- virtual bool **rightOf** (**ICoordinate** \*c)=0  
*Test if this coordinate is to the right of another.*
- virtual void **setE** (double coordEast)=0  
*Set east coordinate.*
- virtual void **setN** (double coordNorth)=0  
*Set north coordinate.*
- virtual void **setH** (double altitude)=0  
*Set altitude.*
- virtual void **shift** (int offsetN, int offsetE)=0  
*Shift coordinate by specified offset.*
- virtual void **divide** (int divisor)=0  
*Divide coordinate by specified divisor.*
- virtual bool **equals** (**ICoordinate** \*c)=0  
*Check if two points match.*
- virtual std::string **toString** ()=0  
*Make string representation.*

---

### Detailed Description

Interface: **Coordinate**.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 35 of file i\_coordinate.h.

---

## Constructor & Destructor Documentation

**virtual sosicon::ICoordinate::~~ICoordinate ()** [*inline*], [*virtual*]

Destructor.

Definition at line 39 of file i\_coordinate.h.

---

## Member Function Documentation

**virtual void sosicon::ICoordinate::divide (int *divisor*)** [*pure virtual*]

Divide coordinate by specified divisor.

Implemented in **sosicon::Coordinate** (*p.62*).

**virtual bool sosicon::ICoordinate::equals (ICoordinate \* *c*)** [*pure virtual*]

Check if two points match.

Implemented in **sosicon::Coordinate** (*p.62*).

**virtual double sosicon::ICoordinate::getE ()** [*pure virtual*]

Get east coordinate.

Implemented in **sosicon::Coordinate** (*p.62*).

**virtual double sosicon::ICoordinate::getN ()** [*pure virtual*]

Get north coordinate.

Implemented in **sosicon::Coordinate** (*p.62*).

**virtual bool sosicon::ICoordinate::leftOf (ICoordinate \* *c*)** [*pure virtual*]

Test if this coordinate is to the left of another.

Implemented in **sosicon::Coordinate** (*p.63*).

**virtual bool sosicon::ICoordinate::rightOf (ICoordinate \* *c*)** [*pure virtual*]

Test if this coordinate is to the right of another.

Implemented in **sosicon::Coordinate** (*p.63*).

**virtual void sosicon::ICoordinate::setE (double *coordEast*) [pure virtual]**

Set east coordinate.

Implemented in **sosicon::Coordinate** (*p.63*).

**virtual void sosicon::ICoordinate::setH (double *altitude*) [pure virtual]**

Set altitude.

Implemented in **sosicon::Coordinate** (*p.63*).

**virtual void sosicon::ICoordinate::setN (double *coordNorth*) [pure virtual]**

Set north coordinate.

Implemented in **sosicon::Coordinate** (*p.63*).

**virtual void sosicon::ICoordinate::shift (int *offsetN*, int *offsetE*) [pure virtual]**

Shift coordinate by specified offset.

Implemented in **sosicon::Coordinate** (*p.63*).

**virtual std::string sosicon::ICoordinate::toString () [pure virtual]**

Make string representation.

Implemented in **sosicon::Coordinate** (*p.63*).

---

**The documentation for this class was generated from the following file:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_coordinate.h`

## sosicon::ILookupTable Class Reference

Interface: Lookup table.

```
#include <i_lookup_table.h>
```

### Public Member Functions

- virtual **~ILookupTable** ()  
*Destructor.*
  - virtual std::string **toString** ()=0  
*Print string representation of current table.*
  - virtual **ISosiElement** \* **get** (std::string key)=0  
*Get SOSI element by ref ID.*
- 

### Detailed Description

Interface: Lookup table.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Interface to a lookup table, implemented by ReferenceLookup class.

Definition at line 40 of file i\_lookup\_table.h.

---

### Constructor & Destructor Documentation

**virtual sosicon::ILookupTable::~~ILookupTable ()** [inline], [virtual]

Destructor.

Definition at line 44 of file i\_lookup\_table.h.

---

### Member Function Documentation

**virtual ISosiElement\* sosicon::ILookupTable::get (std::string key)** [pure virtual]

Get SOSI element by ref ID.

**virtual std::string sosicon::ILookupTable::toString ()** [pure virtual]

Print string representation of current table.

---

**The documentation for this class was generated from the following file:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_lookup_table.h`

## imaxdiv\_t Struct Reference

```
#include <inttypes.h>
```

### Public Attributes

- `intmax_t quot`
  - `intmax_t rem`
- 

### Detailed Description

Definition at line 47 of file `inttypes.h`.

---

### Member Data Documentation

**intmax\_t imaxdiv\_t::quot**

Definition at line 48 of file `inttypes.h`.

**intmax\_t imaxdiv\_t::rem**

Definition at line 49 of file `inttypes.h`.

---

**The documentation for this struct was generated from the following file:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/inttypes.h`

## sosicon::shape::Int16Field Union Reference

16 bit integer / byte field

```
#include <shapefile_types.h>
```

### Public Attributes

- `uint16_t i`
  - `char b [sizeof(uint16_t)]`
- 

### Detailed Description

16 bit integer / byte field

Definition at line 59 of file `shapefile_types.h`.

---

### Member Data Documentation

**`char sosicon::shape::Int16Field::b[sizeof(uint16_t)]`**

Definition at line 61 of file `shapefile_types.h`.

**`uint16_t sosicon::shape::Int16Field::i`**

Definition at line 60 of file `shapefile_types.h`.

---

**The documentation for this union was generated from the following file:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile_types.h`

## **sosicon::shape::Int32Field Union Reference**

32 bit integer / byte field

```
#include <shapefile_types.h>
```

### **Public Attributes**

- `uint32_t i`
  - `char b [sizeof(uint32_t)]`
- 

### **Detailed Description**

32 bit integer / byte field

Definition at line 65 of file `shapefile_types.h`.

---

### **Member Data Documentation**

**`char sosicon::shape::Int32Field::b[sizeof(uint32_t)]`**

Definition at line 67 of file `shapefile_types.h`.

**`uint32_t sosicon::shape::Int32Field::i`**

Definition at line 66 of file `shapefile_types.h`.

---

**The documentation for this union was generated from the following file:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile_types.h`



## sosicon::shape::Int32TField Union Reference

32 bit integer / byte / geom::ShapeType field  
#include <shapefile\_types.h>

### Public Attributes

- uint32\_t **i**
  - char **b** [sizeof(uint32\_t)]
  - ShapeType **t**
- 

### Detailed Description

32 bit integer / byte / geom::ShapeType field  
Definition at line 71 of file shapefile\_types.h.

---

### Member Data Documentation

**char sosicon::shape::Int32TField::b[sizeof(uint32\_t)]**

Definition at line 73 of file shapefile\_types.h.

**uint32\_t sosicon::shape::Int32TField::i**

Definition at line 72 of file shapefile\_types.h.

**ShapeType sosicon::shape::Int32TField::t**

Definition at line 74 of file shapefile\_types.h.

---

**The documentation for this union was generated from the following file:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile\_types.h

## sosicon::shape::Int8Field Union Reference

8 bit integer / byte field

```
#include <shapefile_types.h>
```

### Public Attributes

- `uint8_t i`
  - `char b [sizeof(uint8_t)]`
- 

### Detailed Description

8 bit integer / byte field

Definition at line 53 of file `shapefile_types.h`.

---

### Member Data Documentation

**`char sosicon::shape::Int8Field::b[sizeof(uint8_t)]`**

Definition at line 55 of file `shapefile_types.h`.

**`uint8_t sosicon::shape::Int8Field::i`**

Definition at line 54 of file `shapefile_types.h`.

---

**The documentation for this union was generated from the following file:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile_types.h`

## sosicon::IRectangle Class Reference

Interface: Rectangle.

```
#include <i_rectangle.h>
```

### Public Member Functions

- virtual **~IRectangle** ()  
*Destructor.*
- virtual double **left** ()=0  
*Get left position.*
- virtual void **left** (double val)=0  
*Set left position.*
- virtual double **top** ()=0  
*Get top position.*
- virtual void **top** (double val)=0  
*Set top position.*
- virtual double **right** ()=0  
*Get right position.*
- virtual void **right** (double val)=0  
*Set right position.*
- virtual double **bottom** ()=0  
*Get bottom position.*
- virtual void **bottom** (double val)=0  
*Set bottom position.*

---

### Detailed Description

Interface: Rectangle.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 33 of file i\_rectangle.h.

---

### Constructor & Destructor Documentation

**virtual sosicon::IRectangle::~IRectangle () [inline], [virtual]**

Destructor.

Definition at line 37 of file i\_rectangle.h.

## Member Function Documentation

**virtual double sosicon::IRectangle::bottom () [pure virtual]**

Get bottom position.

Returns the bottom (y1) coordinate of current rectangle.

**Returns:**

Bottom/y1 position.

**virtual void sosicon::IRectangle::bottom (double *val*) [pure virtual]**

Set bottom position.

Sets the bottom (y1) coordinate of current rectangle.

**Parameters:**

<i>val</i>	The new bottom/y1 position.
------------	-----------------------------

**virtual double sosicon::IRectangle::left () [pure virtual]**

Get left position.

Returns the left (x0) coordinate of current rectangle.

**Returns:**

Left/x0 position.

**virtual void sosicon::IRectangle::left (double *val*) [pure virtual]**

Set left position.

Sets the left (x0) coordinate of current rectangle.

**Parameters:**

<i>val</i>	The new left/x0 position.
------------	---------------------------

**virtual double sosicon::IRectangle::right () [pure virtual]**

Get right position.

Returns the right (x1) coordinate of current rectangle.

**Returns:**

Right/x1 position.

**virtual void sosicon::IRectangle::right (double *val*) [pure virtual]**

Set right position.

Sets the right (x1) coordinate of current rectangle.

**Parameters:**

<i>val</i>	The new right/x1 position.
------------	----------------------------

**virtual double sosicon::IRectangle::top () [pure virtual]**

Get top position.

Returns the top (y0) coordinate of current rectangle.

**Returns:**

Top/y0 position.

**virtual void sosicon::IRectangle::top (double *val*) [pure virtual]**

Set top position.

Sets the top (y0) coordinate of current rectangle.

**Parameters:**

<i>val</i>	The new top/y0 position.
------------	--------------------------

---

**The documentation for this class was generated from the following file:**

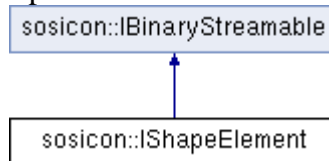
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_rectangle.h

## sosicon::IShapeElement Class Reference

Interface: Shape element.

```
#include <i_shape_element.h>
```

Inheritance diagram for sosicon::IShapeElement:



### Public Member Functions

- virtual `~IShapeElement ()`  
*Destructor.*
- virtual bool **populate** (**ISosiElement** \*sosiElement)=0  
*Create from SOSI element.*
- virtual **ISosiElement** \* **getSosiElement** ()=0  
*Get original SOSI element.*
- virtual void **getMBR** (**IRectangle** &rect)=0  
*Get minimum bounding rectangle.*
- virtual int **getWordSize** ()=0  
*Get element size in 16-bit words.*
- virtual int **getByteSize** ()=0  
*Get element size in bytes.*

---

### Detailed Description

Interface: Shape element.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 37 of file `i_shape_element.h`.

---

### Constructor & Destructor Documentation

**virtual sosicon::IShapeElement::~IShapeElement ()** [`inline`], [`virtual`]

Destructor.

Definition at line 41 of file `i_shape_element.h`.

## Member Function Documentation

**virtual int sosicon::IShapeElement::getByteSize () [pure virtual]**

Get element size in bytes.

Size of current element, in bytes.

**See also:**

**IShapeElement::getWordSize()**

**Returns:**

The vinary size of current element in bytes.

**virtual void sosicon::IShapeElement::getMBR (IRectangle & rect) [pure virtual]**

Get minimum bounding rectangle.

Populates the referenced **IRectangle** implementation with the coordinates for the minimum bounding rectangle (MBR) of current element.

**Parameters:**

<i>Reference</i>	to the <b>IRectangle</b> implementation to receive the coordinates of the minium bounding rectangle.
------------------	--

**virtual ISosiElement\* sosicon::IShapeElement::getSosiElement () [pure virtual]**

Get original SOSI element.

Delivers a pointer to the original SOSI element populating current **IShapeElement**. The pointer is usually stored with current object when **IShapeElement::populate()** is called.

**See also:**

**IShapeElement::populate()**

**Returns:**

Pointer to the source SOSI element.

**virtual int sosicon::IShapeElement::getWordSize () [pure virtual]**

Get element size in 16-bit words.

Size of current element, in 16-bit words. Some parts of the SOSI standard requires size notation on 16-bit words. This is the byte size divided by two.

**See also:**

**IShapeElement::getByteSize()**

**Returns:**

The binary size of current element in 16-bit words. Equals its byte size divided by two.

**virtual bool sosicon::IShapeElement::populate (ISosiElement \* sosiElement) [pure virtual]**

Create from SOSI element.

Takes a SOSI element and creates a shape element from it.

**Parameters:**

<i>sosiElement</i>	Pointer to the SOSI element to be converted to a shape element.
--------------------	---

**Returns:**

The result of the operation.

**Return values:**

<i>true</i>	on success.
<i>false</i>	on failure.

---

**The documentation for this class was generated from the following file:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shape\_element.h

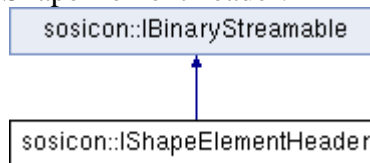


## sosicon::IShapeElementHeader Class Reference

Interface: Shape element header.

```
#include <i_shape_element_header.h>
```

Inheritance diagram for sosicon::IShapeElementHeader:



### Public Member Functions

- `virtual ~IShapeElementHeader ()`  
*Destructor.*

---

### Detailed Description

Interface: Shape element header.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 35 of file `i_shape_element_header.h`.

---

### Constructor & Destructor Documentation

**virtual sosicon::IShapeElementHeader::~IShapeElementHeader () [inline], [virtual]**

Destructor.

Definition at line 39 of file `i_shape_element_header.h`.

---

The documentation for this class was generated from the following file:

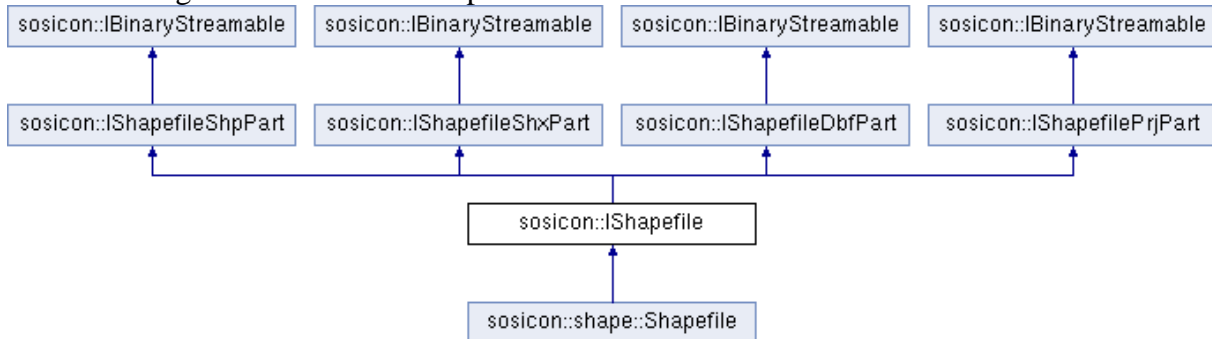
- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shape_element_header.h`

## sosicon::IShapefile Class Reference

Interface: Shapefile.

```
#include <i_shapefile.h>
```

Inheritance diagram for sosicon::IShapefile:



### Public Member Functions

- virtual `~IShapefile()`
- virtual int **build** (`ISosiElement` \*sosiTree, std::string selection, `sosi::ElementType` geomType)=0  
*Build shapefile from SOSI data.*
- virtual void **filterSosiId** (std::vector< std::string > sosiId)=0  
*Set IDs for selected element export.*

---

### Detailed Description

Interface: Shapefile.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 39 of file `i_shapefile.h`.

---

### Constructor & Destructor Documentation

**virtual sosicon::IShapefile::~IShapefile()** [`inline`], [`virtual`]

Definition at line 46 of file `i_shapefile.h`.

---

## Member Function Documentation

**virtual int sosicon::IShapefile::build (ISosiElement \* *sosiTree*, std::string *selection*, sosi::ElementType *geomType*)**[pure virtual]

Build shapefile from SOSI data.

### Parameters:

<i>sosiTree</i>	Root SOSI element. The first-level children of this element will be examined and exported if they are compatible.
<i>selection</i>	SOSI OBJTYPE scheduled for shapefile conversion.
<i>geomType</i>	SOSI element type scheduled for shapefile conversion. Since a shapefile may contain only one geometry type at a time, one must select what element type to extract from the SOSI file.

### Returns:

Number of elements exported.

Implemented in **sosicon::shape::Shapefile** (*p.132*).

**virtual void sosicon::IShapefile::filterSosild (std::vector< std::string > *sosild*)**[pure virtual]

Set IDs for selected element export.

Sets a list of ID flags for elements to be included in the export. All other objects in the source file are ignored.

### Parameters:

<i>sosild</i>	List of the SOSI serials (IDs) of the element(s) to be included in the export.
---------------	--

Implemented in **sosicon::shape::Shapefile** (*p.137*).

---

**The documentation for this class was generated from the following file:**

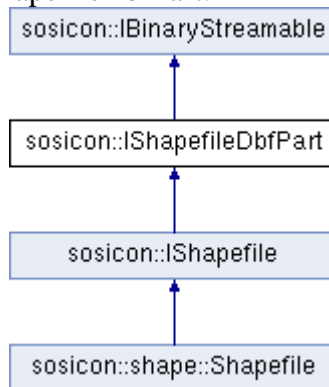
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/**i\_shapefile.h**

## sosicon::IShapefileDbfPart Class Reference

Interface: ShapefileDbfPart.

```
#include <i_shapefile_dbf_part.h>
```

Inheritance diagram for sosicon::IShapefileDbfPart:



### Public Member Functions

- virtual void **writeBinary** (std::ostream &os)  
*Writes binary data to output stream.*
- virtual void **writeDbf** (std::ostream &os)=0

---

### Detailed Description

Interface: ShapefileDbfPart.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 35 of file i\_shapefile\_dbf\_part.h.

---

### Member Function Documentation

**virtual void sosicon::IShapefileDbfPart::writeBinary (std::ostream & os)[inline], [virtual]**

Writes binary data to output stream.

Invoked when the operator<< is used to stream to an std::ostream object. Binary write operation is performed for current impementation here.

#### Parameters:

<i>os</i>	Target stream object.
-----------	-----------------------

Implements **sosicon::IBinaryStreamable** (p.80).

Definition at line 39 of file i\_shapefile\_dbf\_part.h.

**virtual void sosicon::IShapefileDbfPart::writeDbf (std::ostream & os)[pure virtual]**

Implemented in **sosicon::shape::Shapefile** (*p.138*).

---

**The documentation for this class was generated from the following file:**

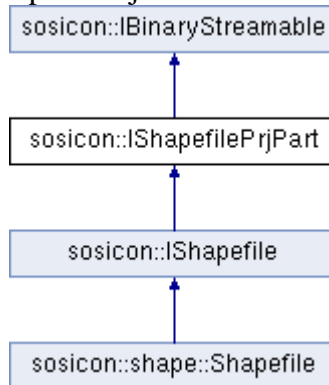
- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_dbf_part.h`

## sosicon::IShapefilePrjPart Class Reference

Interface: ShapefilePrjPart.

```
#include <i_shapefile_prj_part.h>
```

Inheritance diagram for sosicon::IShapefilePrjPart:



### Public Member Functions

- virtual void **writeBinary** (std::ostream &os)  
*Writes binary data to output stream.*
- virtual void **writePrj** (std::ostream &os)=0

---

### Detailed Description

Interface: ShapefilePrjPart.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 35 of file i\_shapefile\_prj\_part.h.

---

### Member Function Documentation

**virtual void sosicon::IShapefilePrjPart::writeBinary (std::ostream & os)[inline], [virtual]**

Writes binary data to output stream.

Invoked when the operator<< is used to stream to an std::ostream object. Binary write operation is performed for current implementation here.

#### Parameters:

<i>os</i>	Target stream object.
-----------	-----------------------

Implements **sosicon::IBinaryStreamable** (p.80).

Definition at line 39 of file i\_shapefile\_prj\_part.h.

**virtual void sosicon::IShapefilePrjPart::writePrj (std::ostream & os)[pure virtual]**

Implemented in **sosicon::shape::Shapefile** (*p.139*).

---

**The documentation for this class was generated from the following file:**

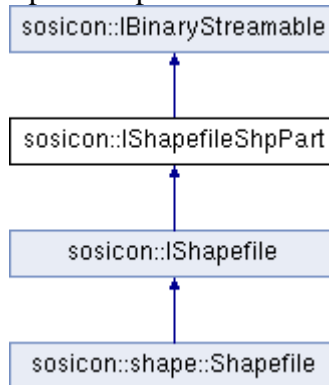
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shapefile\_prj\_part.h

## sosicon::IShapefileShpPart Class Reference

Interface: ShapefileShpPart.

```
#include <i_shapefile_shp_part.h>
```

Inheritance diagram for sosicon::IShapefileShpPart:



### Public Member Functions

- virtual void **writeBinary** (std::ostream &os)  
*Writes binary data to output stream.*
- virtual void **writeShp** (std::ostream &os)=0

---

### Detailed Description

Interface: ShapefileShpPart.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 35 of file i\_shapefile\_shp\_part.h.

---

### Member Function Documentation

**virtual void sosicon::IShapefileShpPart::writeBinary (std::ostream & os)[inline], [virtual]**

Writes binary data to output stream.

Invoked when the operator<< is used to stream to an std::ostream object. Binary write operation is performed for current impementation here.

#### Parameters:

<i>os</i>	Target stream object.
-----------	-----------------------

Implements **sosicon::IBinaryStreamable** (p.80).

Definition at line 39 of file i\_shapefile\_shp\_part.h.



**virtual void sosicon::IShapefileShpPart::writeShp (std::ostream & os)[pure virtual]**

Implemented in **sosicon::shape::Shapefile** (*p.139*).

---

**The documentation for this class was generated from the following file:**

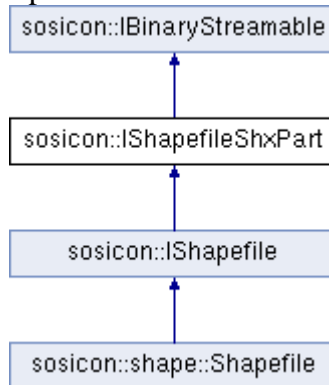
- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_shp_part.h`

## sosicon::IShapefileShxPart Class Reference

Interface: ShapefileShxPart.

```
#include <i_shapefile_shx_part.h>
```

Inheritance diagram for sosicon::IShapefileShxPart:



### Public Member Functions

- virtual void **writeBinary** (std::ostream &os)  
*Writes binary data to output stream.*
- virtual void **writeShx** (std::ostream &os)=0

---

### Detailed Description

Interface: ShapefileShxPart.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 35 of file i\_shapefile\_shx\_part.h.

---

### Member Function Documentation

**virtual void sosicon::IShapefileShxPart::writeBinary** (std::ostream & os)[inline], [virtual]

Writes binary data to output stream.

Invoked when the operator<< is used to stream to an std::ostream object. Binary write operation is performed for current impementation here.

#### Parameters:

<i>os</i>	Target stream object.
-----------	-----------------------

Implements **sosicon::IBinaryStreamable** (p.80).

Definition at line 39 of file i\_shapefile\_shx\_part.h.

**virtual void sosicon::IShapefileShxPart::writeShx (std::ostream & os)[pure virtual]**

Implemented in **sosicon::shape::Shapefile** (*p.139*).

---

**The documentation for this class was generated from the following file:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_shapefile_shx_part.h`

## sosicon::IShapeHeader Class Reference

Interface: Shape element.

```
#include <i_shape_header.h>
```

Inheritance diagram for sosicon::IShapeHeader:



### Public Member Functions

- virtual **~IShapeHeader** ()  
*Destructor.*
- virtual shape::geom::ShapeType **getShapeType** ()=0  
*Return mShapeType.*
- virtual void **setShapeType** (shape::geom::ShapeType shapeType)=0  
*Set mShapeType.*
- virtual void **setFileLength** (int fileLength)=0
- virtual int **getFileLength** ()=0
- virtual void **setBoundingBox** (shape::BoundingBox boundingBox)=0
- virtual shape::BoundingBox **getBoundingBox** ()=0
- virtual int **getByteSize** ()=0  
*Size of header in bytes.*
- virtual int **getWordSize** ()=0  
*Size of header in 16-bit words.*

---

### Detailed Description

Interface: Shape element.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 37 of file `i_shape_header.h`.

---

### Constructor & Destructor Documentation

**virtual sosicon::IShapeHeader::~IShapeHeader () [inline], [virtual]**

Destructor.

Definition at line 41 of file `i_shape_header.h`.

## Member Function Documentation

**virtual shape::BoundingBox sosicon::IShapeHeader::getBoundingBox () [pure virtual]**

**virtual int sosicon::IShapeHeader::getByteSize () [pure virtual]**

Size of header in bytes.

**virtual int sosicon::IShapeHeader::getFileLength () [pure virtual]**

**virtual shape::geom::ShapeType sosicon::IShapeHeader::getShapeType () [pure virtual]**

Return mShapeType.

### See also:

ShapeHeader::mShapeType

### Returns:

The shape type in current file.

**virtual int sosicon::IShapeHeader::getWordSize () [pure virtual]**

Size of header in 16-bit words.

**virtual void sosicon::IShapeHeader::setBoundingBox (shape::BoundingBox *boundingBox*) [pure virtual]**

**virtual void sosicon::IShapeHeader::setFileLength (int *fileLength*) [pure virtual]**

**virtual void sosicon::IShapeHeader::setShapeType (shape::geom::ShapeType *shapeType*) [pure virtual]**

Set mShapeType.

### See also:

ShapeHeader::mShapeType

---

**The documentation for this class was generated from the following file:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shape\_header.h

## sosicon::ISosiElement Class Reference

Interface: SOSI element.

```
#include <i_sosi_element.h>
```

Inheritance diagram for sosicon::ISosiElement:



### Public Member Functions

- virtual **~ISosiElement** ()  
*Destructor.*
- virtual std::vector< **ISosiElement** \* > & **children** ()=0
- virtual void **addChild** (**ISosiElement** \*child)=0
- virtual std::string **getName** ()=0
- virtual bool **getChild** (sosi::SosiElementSearch &src)=0
- virtual std::string **getData** ()=0
- virtual int **getLevel** ()=0
- virtual **sosi::ElementType** **getType** ()=0
- virtual std::string **getObjType** ()=0
- virtual **ISosiElement** \* **getRoot** ()=0
- virtual std::string **getSerial** ()=0
- virtual void **deleteChildren** ()=0
- virtual void **dump** (int indent=0)=0
- virtual **ISosiElement** \* **find** (std::string ref)=0

---

### Detailed Description

Interface: SOSI element.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Represents the generic form of a SOSI element. All SOSI elements must implement this interface. It provides functionality for setting and retrieving field values. The parser writes field values to current SOSI element through this interface.

#### See also:

sosicon::Parser::parseSosiLine()

Data retrieval is provided by one of the **getData()** overloads. Single string data fields are associated with their unique keys, while collections of other data objects, such as address units and cadastral units, are retrieved one-by-one by sequential calls to **getData()**.

Definition at line 51 of file i\_sosi\_element.h.

---

## Constructor & Destructor Documentation

**virtual sosicon::ISosiElement::~~ISosiElement ()** [`inline`], [`virtual`]

Destructor.

Definition at line 55 of file `i_sosi_element.h`.

---

## Member Function Documentation

**virtual void sosicon::ISosiElement::addChild (ISosiElement \* *child*)** [`pure virtual`]

Implemented in `sosicon::sosi::SosiElement` (*p.149*).

**virtual std::vector<ISosiElement\*>& sosicon::ISosiElement::children ()** [`pure virtual`]

Implemented in `sosicon::sosi::SosiElement` (*p.150*).

**virtual void sosicon::ISosiElement::deleteChildren ()** [`pure virtual`]

Implemented in `sosicon::sosi::SosiElement` (*p.150*).

**virtual void sosicon::ISosiElement::dump (int *indent* = 0)** [`pure virtual`]

Implemented in `sosicon::sosi::SosiElement` (*p.150*).

**virtual ISosiElement\* sosicon::ISosiElement::find (std::string *ref*)** [`pure virtual`]

Implemented in `sosicon::sosi::SosiElement` (*p.150*).

**virtual bool sosicon::ISosiElement::getChild (sosi::SosiElementSearch & *src*)** [`pure virtual`]

Implemented in `sosicon::sosi::SosiElement` (*p.150*).

**virtual std::string sosicon::ISosiElement::getData ()** [`pure virtual`]

Implemented in `sosicon::sosi::SosiElement` (*p.150*).

**virtual int sosicon::ISosiElement::getLevel ()** [`pure virtual`]

Implemented in `sosicon::sosi::SosiElement` (*p.150*).

**virtual std::string sosicon::ISosiElement::getName ()** [`pure virtual`]

Implemented in `sosicon::sosi::SosiElement` (*p.151*).

**virtual std::string sosicon::ISosiElement::getObjType () [pure virtual]**

Implemented in `sosicon::sosi::SosiElement` (*p.151*).

**virtual ISosiElement\* sosicon::ISosiElement::getRoot () [pure virtual]**

Implemented in `sosicon::sosi::SosiElement` (*p.151*).

**virtual std::string sosicon::ISosiElement::getSerial () [pure virtual]**

Implemented in `sosicon::sosi::SosiElement` (*p.151*).

**virtual sosi::ElementType sosicon::ISosiElement::getType () [pure virtual]**

Implemented in `sosicon::sosi::SosiElement` (*p.151*).

---

**The documentation for this class was generated from the following file:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_sosi_element.h`

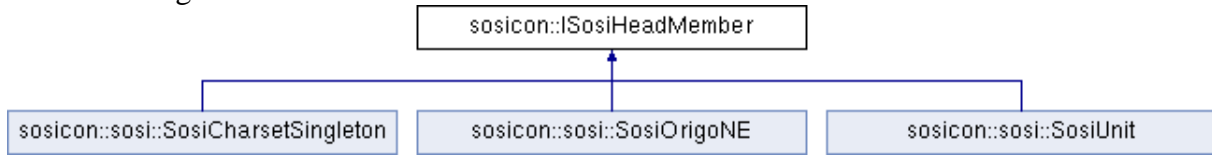


## sosicon::ISosiHeadMember Class Reference

Interface: SOSI header element.

```
#include <i_sosi_head_member.h>
```

Inheritance diagram for sosicon::ISosiHeadMember:



### Public Member Functions

- virtual **~ISosiHeadMember** ()  
*Destructor.*
- virtual void **init** (**ISosiElement** \*e)=0
- virtual bool **initialized** ()=0

---

### Detailed Description

Interface: SOSI header element.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 36 of file i\_sosi\_head\_member.h.

---

### Constructor & Destructor Documentation

**virtual sosicon::ISosiHeadMember::~~ISosiHeadMember ()** [inline], [virtual]

Destructor.

Definition at line 40 of file i\_sosi\_head\_member.h.

---

### Member Function Documentation

**virtual void sosicon::ISosiHeadMember::init** (**ISosiElement** \* e) [pure virtual]

Implemented in **sosicon::sosi::SosiCharsetSingleton** (p.145), **sosicon::sosi::SosiOrigoNE** (p.165), and **sosicon::sosi::SosiUnit** (p.173).

**virtual bool sosicon::ISosiHeadMember::initialized** () [pure virtual]

Implemented in `sosicon::sosi::SosiCharsetSingleton` (p.146), `sosicon::sosi::SosiOrigONE` (p.165), and `sosicon::sosi::SosiUnit` (p.173).

---

**The documentation for this class was generated from the following file:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i_sosi_head_member.h`

## sosicon::EventDispatcher< Event >::Listener Class Reference

```
#include <event_dispatcher.h>
```

### Public Member Functions

- virtual **~Listener** ()
  - virtual void **onEvent** (Event &e, EventDispatcher< Event > &d)=0
- 

### Detailed Description

template<typename Event>

class sosicon::EventDispatcher< Event >::Listener

Definition at line 39 of file event\_dispatcher.h.

---

### Constructor & Destructor Documentation

template<typename Event> virtual sosicon::EventDispatcher< Event >::Listener::~~Listener  
() [inline], [virtual]

Definition at line 42 of file event\_dispatcher.h.

---

### Member Function Documentation

template<typename Event> virtual void sosicon::EventDispatcher< Event >::Listener::onEvent  
(Event & e, EventDispatcher< Event > & d) [pure virtual]

---

The documentation for this class was generated from the following file:

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/event\_dispatcher.h

## sosicon::LogEvent Class Reference

Log event.

```
#include <log_event.h>
```

### Public Member Functions

- **LogEvent** (std::string message, bool update)

### Public Attributes

- std::string **mMessage**
- bool **mUpdate**

---

### Detailed Description

Log event.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

Definition at line 32 of file log\_event.h.

---

### Constructor & Destructor Documentation

**sosicon::LogEvent::LogEvent** (std::string *message*, bool *update*)*[inline]*

Definition at line 35 of file log\_event.h.

---

### Member Data Documentation

**std::string sosicon::LogEvent::mMessage**

Definition at line 38 of file log\_event.h.

**bool sosicon::LogEvent::mUpdate**

Definition at line 39 of file log\_event.h.

---

The documentation for this class was generated from the following file:

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**log\_event.h**



## sosicon::LogEventDispatcher Class Reference

```
#include <log_event.h>
```

Inheritance diagram for sosicon::LogEventDispatcher:



### Additional Inherited Members

---

### Detailed Description

Definition at line 43 of file log\_event.h.

---

The documentation for this class was generated from the following file:

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**log\_event.h**

## sosicon::Logger Class Reference

SOSI logger.  
`#include <logger.h>`

### Public Member Functions

- **Logger** & **operator**<< (std::string v)
- **Logger** & **operator**<< (int v)
- **Logger** & **operator**<< (long v)
- **Logger** & **operator**<< (std::string::size\_type v)
- **Logger** & **operator**<< (**Logger** &(\*func)(**Logger** &))
- void **addEventListener** (LogEventDispatcher::Listener \*listener)
- void **removeEventListener** (LogEventDispatcher::Listener \*listener)

### Private Attributes

- LogEventDispatcher **mLogEventDispatcher**
- std::stringstream **mMsgStream**

---

## Detailed Description

SOSI logger.

#### Author:

Espen Andersen

#### Copyright:

GNU General Public License

User output logger. Redirects to stdin, or a dedicated ILogReceiver implementation.

Definition at line 38 of file logger.h.

---

## Member Function Documentation

**void sosicon::Logger::addEventListener** (LogEventDispatcher::Listener \* *listener*)**[inline]**

Definition at line 51 of file logger.h.

**sosicon::Logger & sosicon::Logger::operator**<< (std::string v)

Definition at line 24 of file logger.cpp.

**sosicon::Logger & sosicon::Logger::operator**<< (int v)

Definition at line 63 of file logger.cpp.

**sosicon::Logger & sosicon::Logger::operator<< (long v)**

Definition at line 71 of file logger.cpp.

**sosicon::Logger & sosicon::Logger::operator<< (std::string::size\_type v)**

Definition at line 55 of file logger.cpp.

**sosicon::Logger & sosicon::Logger::operator<< (Logger &(\*)(Logger &) func)**

Definition at line 79 of file logger.cpp.

**void sosicon::Logger::removeEventListener (LogEventDispatcher::Listener \* listener) [inline]**

Definition at line 52 of file logger.h.

---

## Member Data Documentation

**LogEventDispatcher sosicon::Logger::mLogEventDispatcher [private]**

Definition at line 40 of file logger.h.

**std::stringstream sosicon::Logger::mMsgStream [private]**

Definition at line 41 of file logger.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**logger.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**logger.cpp**



## sosicon::Parser Class Reference

SOSI file parser.

```
#include <parser.h>
```

### Public Member Functions

- **Parser ()**  
*Constructor.*
- **~Parser ()**  
*Destructor.*
- **void complete ()**  
*Flush parsed data.*
- **void dump ()**  
*Debug output.*
- **ISosiElement \* getRootElement ()**  
*Retrieve pointer to root element.*
- **void ragelParseSosiLine (std::string sosiLine)**  
*Main parser routine.*

### Private Member Functions

- **void digestPendingElement ()**  
*Save current SOSI element.*

### Private Attributes

- **std::vector< ISosiElement \* > mElementStack**  
*Working stack.*
- **sosi::SosiElementMap mElementIndex**  
*Index.*
- **sosi::SosiCharsetSingleton \* mCurrentCharset**  
*Current character encoding.*
- **int mPendingElementLevel**  
*SOSI level of element currently in parser.*
- **std::string mPendingElementName**  
*Name of element currently in parser.*
- **std::string mPendingElementSerial**  
*Serial number of element currently in parser.*
- **std::string mPendingElementAttributes**  
*Attribute data of element currently in parser.*

---

## Detailed Description

SOSI file parser.

**Author:**

Espen Andersen

**Copyright:**

GNU General Public License

The file parser. Reads and organizes SOSI file input, preparing the data for conversion and output. This class wraps a Ragel-generated state machine set up to parse SOSI content line-by-line. For more information about the Ragel state machine compiler, visit <http://www.complang.org/ragel/>

Specifically, the function parseSosiLine() is implemented in Ragel. The implementation script is located in parser/parser\_sosi\_line.rl. The file **parser\_ragel.cpp** is generated on the basis of parser/parser.rl during pre-build processing.

**Note:**

Since **parser\_ragel.cpp** is automatically re-generated before each compile, no redacting may take place here.

Any changes will be lost upon compile. Instead, its source script parser/parser.rl has to be edited.

Definition at line 53 of file parser.h.

---

## Constructor & Destructor Documentation

**sosicon::Parser::Parser ()**

Constructor.

Initializes internal iterators and pointers.

Definition at line 21 of file parser.cpp.

**sosicon::Parser::~~Parser ()**

Destructor.

Calls Parser::reset() for final clean-up.

Definition at line 28 of file parser.cpp.

---

## Member Function Documentation

**void sosicon::Parser::complete () [inline]**

Flush parsed data.

Definition at line 123 of file parser.h.

**void sosicon::Parser::digestPendingElement () [private]**

Save current SOSI element.

The parser stores intermediate data in the mPendingElementXXX member variables. When a SOSI element had been fully parsed, this function is called to move the data into the element tree structure before carrying on.

Definition at line 34 of file parser.cpp.

#### **void sosicon::Parser::dump ()**

Debug output.

Definition at line 69 of file parser.cpp.

#### **sosicon::ISosiElement \* sosicon::Parser::getRootElement ()**

Retrieve pointer to root element.

Definition at line 74 of file parser.cpp.

#### **void sosicon::Parser::ragelParseSosiLine (std::string *sosiLine*)**

Main parser routine.

Processes one line from the SOSI file. This function is called repeatedly, consuming the input file line-by-line until EOF.

##### **Note:**

This function is implemented in the ragel script at ragel/parser\_sosi\_line.rl, the c++ file parser\_sosi\_line.cpp is merely generated from the ragel script. Thus, any changes to the implementation must be done in the ragel script, since the c++ file will be automatically overwritten during the pre-build process.

##### **Parameters:**

<i>sosiLine</i>	Current line from the SOSI input file.
-----------------	--

Definition at line 139 of file parser\_ragel.cpp.

---

## **Member Data Documentation**

#### **sosi::SosiCharsetSingleton\* sosicon::Parser::mCurrentCharset [private]**

Current character encoding.

Character encoding of current file in process. Remains undetermined until the TEGNSETT head element is encountered.

Definition at line 74 of file parser.h.

#### **sosi::SosiElementMap sosicon::Parser::mElementIndex [private]**

Index.

Index elements by serial number. Lookup table to resolve SOSI references (REF element).

Definition at line 67 of file parser.h.

#### **std::vector<ISosiElement\*> sosicon::Parser::mElementStack [private]**

Working stack.

Keeps track of element relationship while parsing the SOSI file. The root element is inserted at the front of the stack. When the parser has completed, the stack should contain the root element only.

Definition at line 61 of file parser.h.

**std::string sosicon::Parser::mPendingElementAttributes [private]**

Attribute data of element currently in parser.

Intermediate storage member.

Definition at line 98 of file parser.h.

**int sosicon::Parser::mPendingElementLevel [private]**

SOSI level of element currently in parser.

Intermediate storage member.

Definition at line 80 of file parser.h.

**std::string sosicon::Parser::mPendingElementName [private]**

Name of element currently in parser.

Intermediate storage member.

Definition at line 86 of file parser.h.

**std::string sosicon::Parser::mPendingElementSerial [private]**

Serial number of element currently in parser.

Intermediate storage member.

Definition at line 92 of file parser.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**parser.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**parser.cpp**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**parser\_ragel.cpp**

## sosicon::sosi::ReferenceData Struct Reference

SOSI reference number.

```
#include <sosi_types.h>
```

### Public Attributes

- `std::string serial`  
*The element ID.*
  - `bool reverse`  
*Minus sign = reverse coordinate sequence.*
  - `bool subtract`  
*Parenthesis = subtract shape.*
- 

### Detailed Description

SOSI reference number.

Definition at line 145 of file `sosi_types.h`.

---

### Member Data Documentation

#### `bool sosicon::sosi::ReferenceData::reverse`

Minus sign = reverse coordinate sequence.

Definition at line 147 of file `sosi_types.h`.

#### `std::string sosicon::sosi::ReferenceData::serial`

The element ID.

Definition at line 146 of file `sosi_types.h`.

#### `bool sosicon::sosi::ReferenceData::subtract`

Parenthesis = subtract shape.

Definition at line 148 of file `sosi_types.h`.

---

**The documentation for this struct was generated from the following file:**

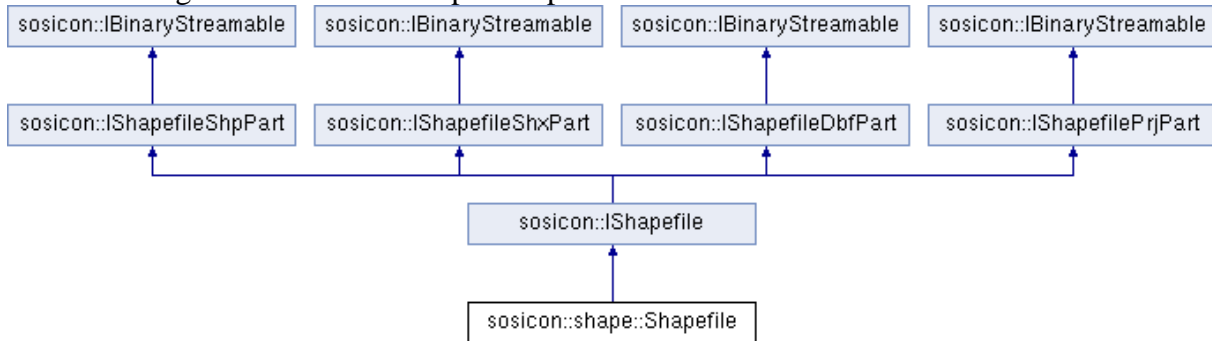
- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_types.h`

## sosicon::shape::Shapefile Class Reference

**Shapefile** implementation.

```
#include <shapefile.h>
```

Inheritance diagram for sosicon::shape::Shapefile:



### Public Member Functions

- **Shapefile ()**  
*Constructor.*
- **virtual ~Shapefile ()**  
*Destructor.*
- **virtual int build (ISosiElement \*sosiTree, std::string objType, sosi::ElementType geomType)**  
*Described in IShapefile.*
- **virtual void filterSosiId (std::vector< std::string > sosiId)**  
*Described in IShapefile.*
- **virtual void writeDbf (std::ostream &os)**  
*Described in IShapefileDbfPart.*
- **virtual void writeShp (std::ostream &os)**  
*Described in IShapefileShpPart.*
- **virtual void writeShx (std::ostream &os)**  
*Described in IShapefileShxPart.*
- **virtual void writePrj (std::ostream &os)**  
*Described in IShapefilePrjPart.*

### Private Member Functions

- **void adjustMasterMbr (double xmin, double ymin, double xmax, double ymax)**  
*Expand MBR to contain Coordinate collection.*
- **void buildShpElement (ISosiElement \*sosi, ShapeType type)**  
*Create SHP element.*
- **void buildShpHeader (ShapeType type)**  
*Populate shape header struct.*
- **void buildShpPoint (CoordinateCollection &cc)**  
*Build shape element: Point.*
- **void buildShpPolygon (CoordinateCollection &cc)**  
*Build shape element: Polygon.*
- **void buildShpPolyLine (CoordinateCollection &cc)**

*Build shape element: PolyLine.*

- void **buildShpRecCoordinate** (int &pos, **CoordinateCollection** &cc)  
*Write first coordinate pair in collection to shapefile buffer.*
- void **buildShpRecCoordinate** (int &pos, **ICoordinate** \*c)  
*Write coordinate pair to shapefile buffer.*
- void **buildShpRecCoordinates** (int &pos, **CoordinateCollection** &cc)  
*Write multiple coordinate pairs to shapefile buffer.*
- void **buildShpRecHeaderCommonPart** (int &pos, int contentLength, **ShapeType** type)  
*Create shapefile record header, common part.*
- void **buildShpRecHeaderExtended** (int &pos, **CoordinateCollection** &cc)  
*Create shapefile record header, extended part.*
- void **buildShpRecHeaderOffsets** (int &pos, **CoordinateCollection** &cc)  
*Create shapefile record header, offsets.*
- void **buildDbf** ()  
*Create DBF file content.*
- void **buildDbfFieldDescriptor** (int &pos)  
*Create DBF field descriptor.*
- void **buildDbfHeader** (int recLen)  
*Create DBF header.*
- void **buildDbfRecordSection** (int &pos, int recLen)  
*Create DBF records.*
- void **buildShx** ()  
*Create SHX file content.*
- void **insertShxOffset** (int contentLen)  
*Append offset value to SHX (index)*
- int **expandShpBuffer** (int byteLen)  
*Expand shp payload buffer.*
- void **extractDbfFields** (**ISosiElement** \*sosi, **DbfRecord** &rec)  
*Recursive func to extract SOSI field data.*
- void **insertDbfRecord** (**ISosiElement** \*sosi)  
*Create and insert DBF record.*
- std::vector< **ICoordinate** \* > **getNormalized** (**sosi::NorthEastList** &neLst)  
*Shapefile polys must have clockwise-ordered vertices.*
- void **saveToDbf** (**DbfRecord** &rec, std::string field, std::string data)  
*Update or insert new DBF field.*

## Private Attributes

- **ISosiElement** \* **mSosiTree**  
*SOSI source.*
- std::vector< std::string > **mFilterSosiId**  
*List of IDs of SOSI elements to be exported, if specified.*
- std::vector< std::string > **mFilterSosiObjTypes**  
*Objtypes of selected elements to be exported, if specified.*
- char **mShpHeader** [100]  
*Main SHP file header.*
- char \* **mShpBuffer**

- *SHP file payload.*
- **int mShpSize**  
*Data length of SHP file buffer.*
- **size\_t mShpBufferSize**  
*Allocated buffer length.*
- **char mShxHeader [100]**  
*Index file header.*
- **char \* mShxBuffer**  
*Index file payload.*
- **size\_t mShxBufferSize**  
*Length of SHX file buffer.*
- **char mDbfHeader [32]**  
*dBase file header*
- **char \* mDbfBuffer**  
*dBase file payload*
- **size\_t mDbfBufferSize**  
*Length of dBase file buffer.*
- **int mRecordNumber**  
*Number of current record in process.*
- **double mXmin**  
*Minimum bounding rectangle, min X.*
- **double mYmin**  
*Minimum bounding rectangle, min Y.*
- **double mXmax**  
*Minimum bounding rectangle, max X.*
- **double mYmax**  
*Minimum bounding rectangle, max Y.*
- **DbfFieldLengths mDbfFieldLengths**  
*Accumulation of DBF fields and their lengths.*
- **DbfRecordSet mDbfRecordSet**  
*All DBF records.*
- **ShxOffsets mShxOffsets**  
*Index file offsets.*

## Static Private Attributes

- **static const int MAX\_BUFFER\_CHUNK\_SIZE**  
*Buffer allocation size.*

---

## Detailed Description

**Shapefile** implementation.

Wraps all ESRI Shape output files (shp, shx, dbf, prj...) in one class.

### Author:

Espen Andersen



## Copyright:

GNU General Public License

Definition at line 54 of file shapefile.h.

---

## Constructor & Destructor Documentation

### **sosicon::shape::Shapefile::Shapefile () [inline]**

Constructor.

Inlined, initializes native members.

Definition at line 367 of file shapefile.h.

### **sosicon::shape::Shapefile::~~Shapefile () [virtual]**

Destructor.

Definition at line 40 of file shapefile.cpp.

---

## Member Function Documentation

### **void sosicon::shape::Shapefile::adjustMasterMbr (double *xMin*, double *yMin*, double *xMax*, double *yMax*) [private]**

Expand MBR to contain **Coordinate** collection.

The minimum bounding rectangle (MBR) for all geometries in current file is stored in members **Shapefile::mXmin**, **Shapefile::mYmin**, **Shapefile::mXmax** and **Shapefile::mYmax**. This method expands the MBR to fit provided coordinates.

#### **Parameters:**

<i>xMin</i>	Minimum X coordinate of geometry to be included i MBR.
<i>yMin</i>	Minimum Y coordinate of geometry to be included i MBR.
<i>xMax</i>	Maximum X coordinate of geometry to be included i MBR.
<i>yMax</i>	Maximum Y coordinate of geometry to be included i MBR.

Definition at line 47 of file shapefile.cpp.

### **int sosicon::shape::Shapefile::build (ISosiElement \* *sosiTree*, std::string *objType*, sosi::ElementType *geomType*) [virtual]**

Described in **IShapefile**.

Implements **sosicon::IShapefile** (*p.102*).

Definition at line 55 of file shapefile.cpp.

### **void sosicon::shape::Shapefile::buildDbf () [private]**

Create DBF file content.

Part of DBF creation. Creates the dBase file content for current shapefile. Populates

**See also:**

**Shapefile::buildDbfHeader**  
**Shapefile::buildDbfFieldDescriptor**  
**Shapefile::buildDbfRecordSection** **Shapefile::mDbfBuffer**.

Definition at line 286 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildDbfFieldDescriptor (int & pos)[private]**

Create DBF field descriptor.

Part of DBF creation. Iterates through individual fields found in current dataset and creates a field descriptor header for the following dBase records.

**See also:**

**Shapefile::buildDbf**  
**Shapefile::buildDbfHeader**  
**Shapefile::buildDbfRecordSection**

**Parameters:**

<i>pos</i>	Reference to an integer holding current position within the shapefile buffer <b>Shapefile::mShpBuffer</b> . The position is updated to reflect the first "free" position after writing to the buffer.
------------	---

Definition at line 320 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildDbfHeader (int recLen)[private]**

Create DBF header.

Part of DBF creation. Creates dBase file header and writes it to **Shapefile::mDbfHeader**.

**See also:**

**Shapefile::buildDbf**  
**Shapefile::buildDbfFieldDescriptor**  
**Shapefile::buildDbfRecordSection**

**Parameters:**

<i>recLen</i>	Length of a single record, in bytes.
---------------	--------------------------------------

Definition at line 351 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildDbfRecordSection (int & pos, int recLen)[private]**

Create DBF records.

Part of DBF creation. Iterates through all records and writes each one to the DBF buffer **Shapefile::mDbfBuffer**.

**See also:**

**Shapefile::buildDbf**  
**Shapefile::buildDbfFieldDescriptor**  
**Shapefile::buildDbfHeader**

**Parameters:**

<i>pos</i>	Reference to an integer holding current position within the shapefile buffer
------------	--

	<b>Shapefile::mShpBuffer</b> . The position is updated to reflect the first "free" position after writing to the buffer.
<i>recLen</i>	Length of a single record, in bytes.

Definition at line 383 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildShpElement (ISosiElement \* *sosi*, ShapeType *type*)[private]**

Create SHP element.

If a shapefile equivalent to current SOSI element exists, this method creates the low-level shape data structure and writes it to the output buffer **Shapefile::mShpBuffer**.

**Parameters:**

<i>sosi</i>	Pointer to SOSI element to be converted to shape.
<i>type</i>	Type of <b>Shapefile</b> geometry equivalent to the SOSI element to be converted.

Definition at line 132 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildShpHeader (ShapeType *type*)[private]**

Populate shape header struct.

Creates master file header for SHP and SHX file parts and writes it to the DBF header buffer **Shapefile::mDbfHeader**.

**Parameters:**

<i>type</i>	The shape type for current file.
-------------	----------------------------------

Definition at line 97 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildShpPoint (CoordinateCollection & *cc*)[private]**

Build shape element: Point.

Inserts a single point into the shapefile buffer.

**Parameters:**

<i>cc</i>	<b>CoordinateCollection</b> containing one or more points. Only the first point in the collection will be handled.
-----------	--

Definition at line 157 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildShpPolygon (CoordinateCollection & *cc*)[private]**

Build shape element: Polygon.

Inserts a polygon into the shapefile buffer.

**Parameters:**

<i>cc</i>	<b>CoordinateCollection</b> containing three or more points, defining the polygon and holes.
-----------	--

Definition at line 179 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildShpPolyLine (CoordinateCollection & cc)[private]**

Build shape element: PolyLine.

Inserts a polyLine into the shapefile buffer.

**Parameters:**

<i>cc</i>	<b>CoordinateCollection</b> containing two or more points, defining the polyLine.
-----------	---

Definition at line 167 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildShpRecCoordinate (int & pos, CoordinateCollection & cc)[private]**

Write first coordinate pair in collection to shapefile buffer.

Build shapefile coordinate from the first coordinate pair in the provided **CoordinateCollection** and update buffer position.

**Parameters:**

<i>pos</i>	Reference to an integer holding current position within the shapefile buffer <b>Shapefile::mShpBuffer</b> . The position is updated to reflect the first "free" position after writing to the buffer.
<i>cc</i>	The coordinate collection from which the first coordinate pair is to be extracted.

Definition at line 191 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildShpRecCoordinate (int & pos, ICoordinate \* c)[private]**

Write coordinate pair to shapefile buffer.

Build shapefile coordinate from the provided coordinate pair and update buffer position.

**Parameters:**

<i>pos</i>	Reference to an integer holding current position within the shapefile buffer <b>Shapefile::mShpBuffer</b> . The position is updated to reflect the first "free" position after writing to the buffer.
<i>c</i>	The coordinate to be written to the buffer.

Definition at line 200 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildShpRecCoordinates (int & pos, CoordinateCollection & cc)[private]**

Write multiple coordinate pairs to shapefile buffer.

Build shapefile coordinate from a collection of coordinate pairs and update buffer position.

**Parameters:**

<i>pos</i>	Reference to an integer holding current position within the shapefile buffer <b>Shapefile::mShpBuffer</b> . The position is updated to reflect the first "free" position after writing to the buffer.
<i>cc</i>	The coordinate collection to be written to the buffer.

Definition at line 208 of file shapefile.cpp.

```
void sosicon::shape::Shapefile::buildShpRecHeaderCommonPart (int & pos, int contentLength, ShapeType type)[private]
```

Create shapefile record header, common part.

The first part of the shapefile record header are common for all geometry types. This method writes the common part to the buffer.

**See also:**

**Shapefile::buildShpRecHeaderExtended**

**Parameters:**

<i>pos</i>	Reference to an integer holding current position within the shapefile buffer <b>Shapefile::mShpBuffer</b> . The position is updated to reflect the first "free" position after writing to the buffer.
<i>contentLength</i>	Length of the record in 16-bit words, record header not included.
<i>type</i>	The shape type for current file.

Definition at line 267 of file shapefile.cpp.

```
void sosicon::shape::Shapefile::buildShpRecHeaderExtended (int & pos, CoordinateCollection & cc)[private]
```

Create shapefile record header, extended part.

For multipoint, polyLine and polygon. This is the second part of the shapefile record header.

**See also:**

**Shapefile::buildShpRecHeaderCommonPart**

**Parameters:**

<i>pos</i>	Reference to an integer holding current position within the shapefile buffer <b>Shapefile::mShpBuffer</b> . The position is updated to reflect the first "free" position after writing to the buffer.
<i>cc</i>	The coordinate collection containing the points for the geometry in current record.

Definition at line 220 of file shapefile.cpp.

```
void sosicon::shape::Shapefile::buildShpRecHeaderOffsets (int & pos, CoordinateCollection & cc)[private]
```

Create shapefile record header, offsets.

The shapefile record header includes a list of offsets to the various parts of the geometry. Applicable to polygons where the main outline is the first part and subsequent parts denotes holes or islands. This method constructs the list of offset values for the multipart geometry and writes it to the shapefile buffer.

**Parameters:**

<i>pos</i>	Reference to an integer holding current position within the shapefile buffer <b>Shapefile::mShpBuffer</b> . The position is updated to reflect the first "free" position after writing to the buffer.
<i>cc</i>	The coordinate collection containing the points for the multi-part geometry in current record.

Definition at line 246 of file shapefile.cpp.

**void sosicon::shape::Shapefile::buildShx () [private]**

Create SHX file content.

Part of SHX index creation. Builds the shapefile index from the **Shapefile::mShxOffsets** entries and writes it to the SHX buffer **Shapefile::mShxBuffer** and the SHX header **Shapefile::mShxHeader**.

**See also:**

**Shapefile::insertShxOffset**

Definition at line 417 of file shapefile.cpp.

**int sosicon::shape::Shapefile::expandShpBuffer (int *byteLen*) [private]**

Expand shp payload buffer.

The shape buffer grows by larger chunks as it is gradually filled up with smaller blocks. For each expansion, the allocation size is doubled, until it reaches MAX\_BUFFER\_CHUNK\_SIZE bytes. This is a tradeoff between execution time and memory consumption. For larger files, there will be relatively few buffer re-allocations and block transfers, in order to save time. For smaller files there will be more frequent re-allocations to save memory.

**Parameters:**

<i>byteLen</i>	The exact length in bytes of the amount of data about to be written to the shapefile buffer. If the current buffer is too small to hold the new block, it will be expanded.
----------------	---

Definition at line 442 of file shapefile.cpp.

**void sosicon::shape::Shapefile::extractDbfFields (ISosiElement \* *sosi*, DbfRecord & *rec*) [private]**

Recursive func to extract SOSI field data.

Traverses the SOSI element, mining the data fields and stores them in the provided record set container.

**See also:**

**Shapefile::insertDbfRecord**

**Parameters:**

<i>sosi</i>	The SOSI element (sub tree) to extract data fields from.
<i>rec</i>	The recordset container to populate with data.

Definition at line 493 of file shapefile.cpp.

**virtual void sosicon::shape::Shapefile::filterSosild (std::vector< std::string > *sosild*) [inline], [virtual]**

Described in **IShapefile**.

Implements **sosicon::IShapefile** (p.102).

Definition at line 389 of file shapefile.h.

**std::vector<ICoordinate\*> sosicon::shape::Shapefile::getNormalized (sosi::NorthEastList & neLst)[private]**

**Shapefile** polys must have clockwise-ordered vertices.

This is one of the core functions for handling polygons. Tests the direction for SOSI coordinates, and reverses them if they are in the wrong order with respect to the **Shapefile** format epscification.

**Note:**

The direction of the vertices in a SOSI polygon is not significant, whist in a **Shapefile** polygon, it is crucial. The vertices in the outer polygon should always be ordered in a clockwise direction, while the holes or islands must be ordered in a counter-clockwise direction.

**Parameters:**

<i>neLst</i>	List of SOSI NorthEast elements describing current multipart gemoetry (polygon with holes/islands).
--------------	---

**void sosicon::shape::Shapefile::insertDbfRecord (ISosiElement \* sosi)[private]**

Create and insert DBF record.

Prepares dBase record for current SOSI element. Creates the two mandatory fields "SOSI\_ID" and "TYPE", before it calls Shapefil::extractDbfFields to retrieve the other data fields. The record is then inserted into the **Shapefile::mDbfRecordSet** member.

**See also:**

Shapefil::extractDbfFields

**Parameters:**

<i>sosi</i>	The SOSI element (sub tree) to extract data fields from.
-------------	--

Definition at line 511 of file shapefile.cpp.

**void sosicon::shape::Shapefile::insertShxOffset (int contentLen)[private]**

Append offset value to SHX (index)

For each shapefile record, it's offset within the main file is pushed to the **Shapefile::mShxOffsets** vector.

**Parameters:**

<i>contentLen</i>	Length of the shapefile record content, in 16-bit words, record header not included.
-------------------	--

Definition at line 520 of file shapefile.cpp.

**void sosicon::shape::Shapefile::saveToDbf (DbfRecord & rec, std::string field, std::string data)[private]**

Update or insert new DBF field.

Appends or updates data for the DFB record, updating list of field names and lengths.

Definition at line 528 of file shapefile.cpp.

**void sosicon::shape::Shapefile::writeDbf (std::ostream & os)[virtual]**

Described in **IShapefileDbfPart**.

Implements **sosicon::IShapefileDbfPart** (*p.104*).

Definition at line 554 of file shapefile.cpp.

**void sosicon::shape::Shapefile::writePrj (std::ostream & os)[virtual]**

Described in **IShapefilePrjPart**.

Implements **sosicon::IShapefilePrjPart** (*p.106*).

Definition at line 560 of file shapefile.cpp.

**void sosicon::shape::Shapefile::writeShp (std::ostream & os)[virtual]**

Described in **IShapefileShpPart**.

Implements **sosicon::IShapefileShpPart** (*p.108*).

Definition at line 542 of file shapefile.cpp.

**void sosicon::shape::Shapefile::writeShx (std::ostream & os)[virtual]**

Described in **IShapefileShxPart**.

Implements **sosicon::IShapefileShxPart** (*p.110*).

Definition at line 548 of file shapefile.cpp.

---

## Member Data Documentation

**const int sosicon::shape::Shapefile::MAX\_BUFFER\_CHUNK\_SIZE [static], [private]**

Buffer allocation size.

To speed things up, heap allocations are done in increasingly large chunks. This parameters defines how many bytes to request for the largest buffer expansion.

Definition at line 62 of file shapefile.h.

**char\* sosicon::shape::Shapefile::mDbfBuffer [private]**

dBase file payload

Definition at line 79 of file shapefile.h.

**size\_t sosicon::shape::Shapefile::mDbfBufferSize [private]**

Length of dBase file buffer.

Definition at line 80 of file shapefile.h.



**DbfFieldLengths** **sosicon::shape::Shapefile::mDbfFieldLengths** [private]

Accumulation of DBF fields and their lengths.

Definition at line 89 of file shapefile.h.

**char** **sosicon::shape::Shapefile::mDbfHeader**[32] [private]

dBase file header

Definition at line 78 of file shapefile.h.

**DbfRecordSet** **sosicon::shape::Shapefile::mDbfRecordSet** [private]

All DBF records.

Definition at line 90 of file shapefile.h.

**std::vector<std::string>** **sosicon::shape::Shapefile::mFilterSosiId** [private]

List of IDs of SOSI elements to be exported, if specified.

Definition at line 66 of file shapefile.h.

**std::vector<std::string>** **sosicon::shape::Shapefile::mFilterSosiObjTypes** [private]

Objtypes of selected elements to be exported, if specified.

Definition at line 67 of file shapefile.h.

**int** **sosicon::shape::Shapefile::mRecordNumber** [private]

Number of current record in process.

Definition at line 82 of file shapefile.h.

**char\*** **sosicon::shape::Shapefile::mShpBuffer** [private]

SHP file payload.

Definition at line 70 of file shapefile.h.

**size\_t** **sosicon::shape::Shapefile::mShpBufferSize** [private]

Allocated buffer length.

Definition at line 72 of file shapefile.h.

**char** **sosicon::shape::Shapefile::mShpHeader**[100] [private]

Main SHP file header.

Definition at line 69 of file shapefile.h.

**int sosicon::shape::Shapefile::mShpSize [private]**

Data length of SHP file buffer.

Definition at line 71 of file shapefile.h.

**char\* sosicon::shape::Shapefile::mShxBuffer [private]**

Index file payload.

Definition at line 75 of file shapefile.h.

**size\_t sosicon::shape::Shapefile::mShxBufferSize [private]**

Length of SHX file buffer.

Definition at line 76 of file shapefile.h.

**char sosicon::shape::Shapefile::mShxHeader[100] [private]**

Index file header.

Definition at line 74 of file shapefile.h.

**ShxOffsets sosicon::shape::Shapefile::mShxOffsets [private]**

Index file offsets.

Definition at line 91 of file shapefile.h.

**ISosiElement\* sosicon::shape::Shapefile::mSosiTree [private]**

SOSI source.

Definition at line 64 of file shapefile.h.

**double sosicon::shape::Shapefile::mXmax [private]**

Minimum bounding rectangle, max X.

Definition at line 86 of file shapefile.h.

**double sosicon::shape::Shapefile::mXmin [private]**

Minimum bounding rectangle, min X.

Definition at line 84 of file shapefile.h.

**double sosicon::shape::Shapefile::mYmax [private]**

Minimum bounding rectangle, max Y.  
Definition at line 87 of file shapefile.h.

**double sosicon::shape::Shapefile::mYmin [private]**

Minimum bounding rectangle, min Y.  
Definition at line 85 of file shapefile.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/**shapefile.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/**shapefile.cpp**

## sosicon::shape::ShxIndex Struct Reference

```
#include <shapefile_types.h>
```

### Public Attributes

- **Int32Field** offset
  - **Int32Field** length
- 

### Detailed Description

Definition at line 83 of file shapefile\_types.h.

---

### Member Data Documentation

#### **Int32Field** sosicon::shape::ShxIndex::length

Definition at line 85 of file shapefile\_types.h.

#### **Int32Field** sosicon::shape::ShxIndex::offset

Definition at line 84 of file shapefile\_types.h.

---

The documentation for this struct was generated from the following file:

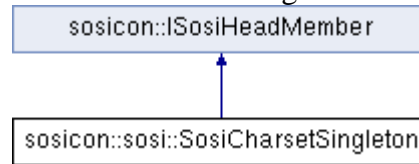
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile\_types.h

## sosicon::sosi::SosiCharsetSingleton Class Reference

SOSI Character set.

```
#include <sosi_charset_singleton.h>
```

Inheritance diagram for sosicon::sosi::SosiCharsetSingleton:



### Public Member Functions

- virtual **~SosiCharsetSingleton** ()  
*Destructor.*
- **SosiCharsetSingleton** (ISosiElement \*e)  
*Construct new SOSI Charset element.*
- **Charset** **getEncoding** ()
- std::string **getEncodingName** ()
- virtual void **init** (ISosiElement \*e)  
*Initialize SOSI Unit element.*
- virtual bool **initialized** ()
- std::string **toIso8859\_1** (const std::string &str)  
*Convert string to ISO8859-1 (default Ragel charset)*

### Static Public Member Functions

- static **SosiCharsetSingleton** \* **getInstance** ()

### Private Member Functions

- **SosiCharsetSingleton** ()  
*Construct new SOSI Charset element.*

### Static Private Member Functions

- static std::string **utf8ToIso8859\_1** (const char \*in)  
*Quick and dirty conversion from UTF-8 to ISO8859-10.*

### Private Attributes

- **ISosiElement** \* **mSosiElement**
- bool **mInitialized**
- **Charset** **mCharset**  
*Type of character set.*
- std::string **mCharsetName**  
*Name of character set.*

### Static Private Attributes

- static **SosiCharsetSingleton** \* **mInstance**

## Detailed Description

SOSI Character set.

Implements SOSI character set, as given via the TEGNSETT element.

Definition at line 114 of file `sosi_charset_singleton.h`.

---

## Constructor & Destructor Documentation

**`sosicon::sosi::SosiCharsetSingleton::SosiCharsetSingleton () [private]`**

Construct new SOSI Charset element.

Declared private because it's a singleton.

Definition at line 23 of file `sosi_charset_singleton.cpp`.

**`virtual sosicon::sosi::SosiCharsetSingleton::~SosiCharsetSingleton () [inline], [virtual]`**

Destructor.

Definition at line 150 of file `sosi_charset_singleton.h`.

**`sosicon::sosi::SosiCharsetSingleton::SosiCharsetSingleton (ISosiElement * e) [inline]`**

Construct new SOSI Charset element.

Definition at line 153 of file `sosi_charset_singleton.h`.

---

## Member Function Documentation

**`Charset sosicon::sosi::SosiCharsetSingleton::getEncoding () [inline]`**

Definition at line 155 of file `sosi_charset_singleton.h`.

**`std::string sosicon::sosi::SosiCharsetSingleton::getEncodingName () [inline]`**

Definition at line 157 of file `sosi_charset_singleton.h`.

**`static SosiCharsetSingleton* sosicon::sosi::SosiCharsetSingleton::getInstance () [inline], [static]`**

Definition at line 142 of file `sosi_charset_singleton.h`.

**`void sosicon::sosi::SosiCharsetSingleton::init (ISosiElement * e) [virtual]`**

Initialize SOSI Unit element.

Implements **sosicon::ISosiHeadMember** (*p.116*).

Definition at line 30 of file `sosi_charset_singleton.cpp`.

**virtual bool sosicon::sosi::SosiCharsetSingleton::initialized () [inline], [virtual]**

Implements **sosicon::ISosiHeadMember** (*p.116*).

Definition at line 162 of file `sosi_charset_singleton.h`.

**std::string sosicon::sosi::SosiCharsetSingleton::tolso8859\_1 (const std::string & str)**

Convert string to ISO8859-1 (default Regel charset)

Definition at line 45 of file `sosi_charset_singleton.cpp`.

**std::string sosicon::sosi::SosiCharsetSingleton::utf8Tolso8859\_1 (const char \* in) [static], [private]**

Quick and dirty conversion from UTF-8 to ISO8859-10.

Invalid characters are dropped. Sorry.

Definition at line 75 of file `sosi_charset_singleton.cpp`.

---

## Member Data Documentation

**Charset sosicon::sosi::SosiCharsetSingleton::mCharset [private]**

Type of character set.

Definition at line 123 of file `sosi_charset_singleton.h`.

**std::string sosicon::sosi::SosiCharsetSingleton::mCharsetName [private]**

Name of character set.

Definition at line 126 of file `sosi_charset_singleton.h`.

**bool sosicon::sosi::SosiCharsetSingleton::mInitialized [private]**

Definition at line 120 of file `sosi_charset_singleton.h`.

**sosicon::sosi::SosiCharsetSingleton \* sosicon::sosi::SosiCharsetSingleton::mInstance [static], [private]**

Definition at line 116 of file `sosi_charset_singleton.h`.

**ISosiElement\* sosicon::sosi::SosiCharsetSingleton::mSosiElement [private]**

Definition at line 118 of file `sosi_charset_singleton.h`.

---

**The documentation for this class was generated from the following files:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_charset_singleton.h`
- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_charset_singleton.cpp`



## sosicon::sosi::SosiElement Class Reference

Basic SOSI element.

```
#include <sosi_element.h>
```

Inheritance diagram for sosicon::sosi::SosiElement:



### Public Member Functions

- **SosiElement** (std::string name, std::string serial, std::string data, int level, **ISosiElement** \*root, **SosiElementMap** &index)  
*Construct new SOSI element.*
- virtual void **addChild** (**ISosiElement** \*child)  
*Insert children element.*
- virtual void **deleteChildren** ()  
*Recursively deletes all children.*
- virtual void **dump** (int indent=0)  
*Debug function.*
- virtual **ISosiElement** \* **find** (std::string ref)  
*Find element by reference.*
- std::vector< **ISosiElement** \* > & **children** ()
- virtual bool **getChild** (**SosiElementSearch** &src)  
*Get next child in list.*
- virtual std::string **getData** ()  
*Get unparsed element data.*
- virtual int **getLevel** ()  
*Get nesting level of current element.*
- virtual std::string **getObjType** ()  
*Get ObjType of current element.*
- virtual std::string **getName** ()  
*Get name of current element.*
- virtual **ISosiElement** \* **getRoot** ()  
*Get root element.*
- virtual std::string **getSerial** ()  
*Get serial number (ID) of current element.*
- virtual **ElementType** **getType** ()  
*Get ElementType of current element.*

### Private Member Functions

- virtual bool **nextChild** (**SosiElementSearch** &src)  
*Increment to next child in list.*

## Private Attributes

- **SosiTranslationTable mTranslation**  
*SOSI string translations.*
- **std::string mData**  
*Current element's data content.*
- **SosiChildrenList mChildren**  
*List of children elements.*
- **int mLevel**  
*Current element's nesting level.*
- **std::string mName**  
*Current element's name.*
- **ElementType mType**  
*Current element's geometric type.*
- **ObjType mObjType**  
*Current element's objtype.*
- **std::string mObjTypeStr**  
*Current element's objtype.*
- **std::string mSerial**  
*Current element's serial number if provided.*
- **ISosiElement \* mRoot**  
*Pointer to root element.*
- **SosiElementMap & mIndex**  
*Reference to parser's lookup table.*

---

## Detailed Description

Basic SOSI element.

Implements basic characteristics of a SOSI element.

Definition at line 72 of file `sosi_element.h`.

---

## Constructor & Destructor Documentation

**sosicon::sosi::SosiElement::SosiElement** (std::string *name*, std::string *serial*, std::string *data*, int *level*, ISosiElement \* *root*, SosiElementMap & *index*)

Construct new SOSI element.

Definition at line 21 of file `sosi_element.cpp`.

---

## Member Function Documentation

**void sosicon::sosi::SosiElement::addChild** (ISosiElement \* *child*)`[virtual]`

Insert children element.

Implements **sosicon::ISosiElement** (*p.114*).

Definition at line 35 of file `sosi_element.cpp`.

**std::vector<ISosiElement\*>& sosicon::sosi::SosiElement::children ()** [*inline*], [*virtual*]

Implements **sosicon::ISosiElement** (*p.114*).

Definition at line 127 of file `sosi_element.h`.

**void sosicon::sosi::SosiElement::deleteChildren ()** [*virtual*]

Recursively deletes all children.

Implements **sosicon::ISosiElement** (*p.114*).

Definition at line 44 of file `sosi_element.cpp`.

**void sosicon::sosi::SosiElement::dump (int *indent* = 0)** [*virtual*]

Debug function.

Implements **sosicon::ISosiElement** (*p.114*).

Definition at line 52 of file `sosi_element.cpp`.

**sosicon::ISosiElement \* sosicon::sosi::SosiElement::find (std::string *ref*)** [*virtual*]

Find element by reference.

Implements **sosicon::ISosiElement** (*p.114*).

Definition at line 62 of file `sosi_element.cpp`.

**bool sosicon::sosi::SosiElement::getChild (SosiElementSearch & *src*)** [*virtual*]

Get next child in list.

Always pass a null pointer to start iterating through the children list. The referenced pointer will point to the next child in list when the function returns. If the end of the list is reached, the function returns false.

Implements **sosicon::ISosiElement** (*p.114*).

Definition at line 100 of file `sosi_element.cpp`.

**virtual std::string sosicon::sosi::SosiElement::getData ()** [*inline*], [*virtual*]

Get unparsed element data.

Implements **sosicon::ISosiElement** (*p.114*).

Definition at line 138 of file `sosi_element.h`.

**virtual int sosicon::sosi::SosiElement::getLevel ()** [*inline*], [*virtual*]

Get nesting level of current element.

Implements **sosicon::ISosiElement** (p.114).

Definition at line 141 of file sosi\_element.h.

**std::string sosicon::sosi::SosiElement::getName () [virtual]**

Get name of current element.

Implements **sosicon::ISosiElement** (p.114).

Definition at line 94 of file sosi\_element.cpp.

**virtual std::string sosicon::sosi::SosiElement::getObjType () [inline], [virtual]**

Get ObjType of current element.

Implements **sosicon::ISosiElement** (p.115).

Definition at line 144 of file sosi\_element.h.

**virtual ISosiElement\* sosicon::sosi::SosiElement::getRoot () [inline], [virtual]**

Get root element.

Implements **sosicon::ISosiElement** (p.115).

Definition at line 150 of file sosi\_element.h.

**virtual std::string sosicon::sosi::SosiElement::getSerial () [inline], [virtual]**

Get serial number (ID) of current element.

Implements **sosicon::ISosiElement** (p.115).

Definition at line 153 of file sosi\_element.h.

**virtual ElementType sosicon::sosi::SosiElement::getType () [inline], [virtual]**

Get ElementType of current element.

Implements **sosicon::ISosiElement** (p.115).

Definition at line 156 of file sosi\_element.h.

**bool sosicon::sosi::SosiElement::nextChild (SosiElementSearch & src) [private], [virtual]**

Increment to next child in list.

Definition at line 74 of file sosi\_element.cpp.

## Member Data Documentation

### **SosiChildrenList** **sosicon::sosi::SosiElement::mChildren** [private]

List of children elements.

Definition at line 81 of file sosi\_element.h.

### **std::string** **sosicon::sosi::SosiElement::mData** [private]

Current element's data content.

Definition at line 78 of file sosi\_element.h.

### **SosiElementMap&** **sosicon::sosi::SosiElement::mIndex** [private]

Reference to parser's lookup table.

Definition at line 105 of file sosi\_element.h.

### **int** **sosicon::sosi::SosiElement::mLevel** [private]

Current element's nesting level.

Definition at line 84 of file sosi\_element.h.

### **std::string** **sosicon::sosi::SosiElement::mName** [private]

Current element's name.

Definition at line 87 of file sosi\_element.h.

### **ObjType** **sosicon::sosi::SosiElement::mObjType** [private]

Current element's objtype.

Definition at line 93 of file sosi\_element.h.

### **std::string** **sosicon::sosi::SosiElement::mObjTypeStr** [private]

Current element's objtype.

Definition at line 96 of file sosi\_element.h.

### **ISosiElement\*** **sosicon::sosi::SosiElement::mRoot** [private]

Pointer to root element.

Definition at line 102 of file sosi\_element.h.

**std::string sosicon::sosi::SosiElement::mSerial** [private]

Current element's serial number if provided.

Definition at line 99 of file sosi\_element.h.

**SosiTranslationTable sosicon::sosi::SosiElement::mTranslation** [private]

SOSI string translations.

Definition at line 75 of file sosi\_element.h.

**ElementType sosicon::sosi::SosiElement::mType** [private]

Current element's geometric type.

Definition at line 90 of file sosi\_element.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/**sosi\_element.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/**sosi\_element.cpp**

## sosicon::sosi::SosiElementSearch Class Reference

```
#include <sosi_element_search.h>
```

### Public Member Functions

- **SosiElementSearch** ()
- **SosiElementSearch** (sosi::ElementType filter)
- **SosiElementSearch** (std::vector< sosi::ElementType > &filterList)
- SosiChildrenList::size\_type **index** ()
- SosiChildrenList::size\_type **index** (SosiChildrenList::size\_type i)
- **ISosiElement** \* **element** ()
- **ISosiElement** \* **element** (ISosiElement \*e)
- sosi::ElementType **type** ()
- sosi::ElementType **type** (sosi::ElementType t)
- std::vector< sosi::ElementType > & **types** ()
- std::vector< sosi::ElementType > & **types** (std::vector< sosi::ElementType > &t)
- bool **matchTypes** ()
- void **next** ()

### Private Attributes

- SosiChildrenList::size\_type **mIndex**
- **ISosiElement** \* **mSosiElement**
- std::vector< sosi::ElementType > **mElementTypes**

---

## Detailed Description

Definition at line 46 of file sosi\_element\_search.h.

---

## Constructor & Destructor Documentation

**sosicon::sosi::SosiElementSearch::SosiElementSearch** () [*inline*]

Definition at line 51 of file sosi\_element\_search.h.

**sosicon::sosi::SosiElementSearch::SosiElementSearch** (sosi::ElementType *filter*) [*inline*]

Definition at line 52 of file sosi\_element\_search.h.

**sosicon::sosi::SosiElementSearch::SosiElementSearch** (std::vector< sosi::ElementType > &*filterList*) [*inline*]

Definition at line 53 of file sosi\_element\_search.h.

---

## Member Function Documentation

**ISosiElement\* sosicon::sosi::SosiElementSearch::element () [inline]**

Definition at line 56 of file sosi\_element\_search.h.

**ISosiElement\* sosicon::sosi::SosiElementSearch::element (ISosiElement \* e) [inline]**

Definition at line 57 of file sosi\_element\_search.h.

**SosiChildrenList::size\_type sosicon::sosi::SosiElementSearch::index () [inline]**

Definition at line 54 of file sosi\_element\_search.h.

**SosiChildrenList::size\_type sosicon::sosi::SosiElementSearch::index (SosiChildrenList::size\_type i) [inline]**

Definition at line 55 of file sosi\_element\_search.h.

**bool sosicon::sosi::SosiElementSearch::matchTypes ()**

Definition at line 22 of file sosi\_element\_search.cpp.

**void sosicon::sosi::SosiElementSearch::next () [inline]**

Definition at line 63 of file sosi\_element\_search.h.

**sosi::ElementType sosicon::sosi::SosiElementSearch::type () [inline]**

Definition at line 58 of file sosi\_element\_search.h.

**sosi::ElementType sosicon::sosi::SosiElementSearch::type (sosi::ElementType t) [inline]**

Definition at line 59 of file sosi\_element\_search.h.

**std::vector<sosi::ElementType>& sosicon::sosi::SosiElementSearch::types () [inline]**

Definition at line 60 of file sosi\_element\_search.h.

**std::vector<sosi::ElementType>& sosicon::sosi::SosiElementSearch::types (std::vector<sosi::ElementType> & t) [inline]**

Definition at line 61 of file sosi\_element\_search.h.



## Member Data Documentation

**std::vector<sosi::ElementType> sosicon::sosi::SosiElementSearch::mElementTypes** [private]

Definition at line 49 of file sosi\_element\_search.h.

**SosiChildrenList::size\_type sosicon::sosi::SosiElementSearch::mIndex** [private]

Definition at line 47 of file sosi\_element\_search.h.

**ISosiElement\* sosicon::sosi::SosiElementSearch::mSosiElement** [private]

Definition at line 48 of file sosi\_element\_search.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_element\_search.h
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_element\_search.cpp

## sosicon::sosi::SosiJunctionPoint Class Reference

SOSI Junction point.

```
#include <sosi_junction_point.h>
```

### Public Member Functions

- **SosiJunctionPoint (ISosiElement \*e)**  
*Construct new SOSI junction point element.*
- **~SosiJunctionPoint ()**  
*Destructor.*

### Private Attributes

- **ISosiElement \* mSosiElement**

---

### Detailed Description

SOSI Junction point.

Implements SOSI junction point layer specification, as given via the KP element. Norwegian: Knutepunktslag.

Definition at line 43 of file `sosi_junction_point.h`.

---

### Constructor & Destructor Documentation

**sosicon::sosi::SosiJunctionPoint::SosiJunctionPoint (ISosiElement \* e)[inline]**

Construct new SOSI junction point element.

Definition at line 50 of file `sosi_junction_point.h`.

**sosicon::sosi::SosiJunctionPoint::~~SosiJunctionPoint ()**

Destructor.

---

### Member Data Documentation

**ISosiElement\* sosicon::sosi::SosiJunctionPoint::mSosiElement [private]**

Definition at line 45 of file `sosi_junction_point.h`.

---

**The documentation for this class was generated from the following file:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_junction_point.h`



## sosicon::sosi::SosiNorthEast Class Reference

SOSI North-east element.

```
#include <sosi_north_east.h>
```

### Public Member Functions

- void **append** (double n, double e)
- void **append** (double n, double e, double h)
- void **append** (std::string n, std::string e)
- void **append** (std::string n, std::string e, std::string h)  
*Frees allocated memory.*
- void **free** ()
- **SosiNorthEast** (**ISosiElement** \*e)  
*Construct new SOSI north-east element.*
- virtual ~**SosiNorthEast** ()  
*Destructor.*
- void **dump** ()  
*Debug.*
- void **expandBoundingBox** (double &minX, double &minY, double &maxX, double &maxY)
- **ICoordinate** \* **front** ()
- **ICoordinate** \* **back** ()
- bool **getNext** (**ICoordinate** \*&coord)
- int **getNumPoints** ()
- void **reverse** ()  
*Reverse polygon (point order)*
- **SosiNorthEast** & **operator+=** (**SosiOrigoNE** &origo)
- **SosiNorthEast** & **operator/=** (**SosiUnit** &unit)

### Private Member Functions

- void **ragelParseCoordinatesNe** (std::string data)  
*Populate mCoordinates.*
- void **ragelParseCoordinatesNeh** (std::string data)
- void **initHeadMember** (**ISosiHeadMember** &headMember, **ElementType** type)

### Private Attributes

- **ISosiElement** \* **mSosiElement**
- **CoordinateList** **mCoordinates**
- **CoordinateList::iterator** **mCoordinatesIterator**
- double **mMinX**
- double **mMinY**
- double **mMaxX**
- double **mMaxY**

### Static Private Attributes

- static **SosiOrigoNE** **mOrigo**
- static **SosiUnit** **mUnit**

## Detailed Description

SOSI North-east element.

Implements SOSI north east element, as given via the NØ element.

Definition at line 50 of file sosi\_north\_east.h.

---

## Constructor & Destructor Documentation

**sosicon::sosi::SosiNorthEast::SosiNorthEast (ISosiElement \* e)**

Construct new SOSI north-east element.

Definition at line 33 of file sosi\_north\_east.cpp.

**sosicon::sosi::SosiNorthEast::~~SosiNorthEast () [virtual]**

Destructor.

Definition at line 54 of file sosi\_north\_east.cpp.

---

## Member Function Documentation

**void sosicon::sosi::SosiNorthEast::append (double n, double e)**

Definition at line 89 of file sosi\_north\_east.cpp.

**void sosicon::sosi::SosiNorthEast::append (double n, double e, double h)**

Definition at line 94 of file sosi\_north\_east.cpp.

**void sosicon::sosi::SosiNorthEast::append (std::string n, std::string e)**

Definition at line 57 of file sosi\_north\_east.cpp.

**void sosicon::sosi::SosiNorthEast::append (std::string n, std::string e, std::string h)**

Frees allocated memory.

Definition at line 68 of file sosi\_north\_east.cpp.

**ISosiNorthEast\* sosicon::sosi::SosiNorthEast::back () [inline]**

Definition at line 96 of file sosi\_north\_east.h.

**void sosicon::sosi::SosiNorthEast::dump ()**

Debug.

Definition at line 132 of file sosi\_north\_east.cpp.

**void sosicon::sosi::SosiNorthEast::expandBoundingBox (double & *minX*, double & *minY*, double & *maxX*, double & *maxY*)**

Definition at line 139 of file sosi\_north\_east.cpp.

**void sosicon::sosi::SosiNorthEast::free ()**

Definition at line 81 of file sosi\_north\_east.cpp.

**ICoordinate\* sosicon::sosi::SosiNorthEast::front () [inline]**

Definition at line 94 of file sosi\_north\_east.h.

**bool sosicon::sosi::SosiNorthEast::getNext (ICoordinate \*& *coord*)**

Definition at line 147 of file sosi\_north\_east.cpp.

**int sosicon::sosi::SosiNorthEast::getNumPoints () [inline]**

Definition at line 102 of file sosi\_north\_east.h.

**void sosicon::sosi::SosiNorthEast::initHeadMember (ISosiHeadMember & *headMember*, ElementType *type*) [private]**

Definition at line 116 of file sosi\_north\_east.cpp.

**sosicon::sosi::SosiNorthEast & sosicon::sosi::SosiNorthEast::operator+= (SosiOrigoNE & *origo*)**

Definition at line 163 of file sosi\_north\_east.cpp.

**sosicon::sosi::SosiNorthEast & sosicon::sosi::SosiNorthEast::operator/= (SosiUnit & *unit*)**

Definition at line 179 of file sosi\_north\_east.cpp.

**void sosicon::sosi::SosiNorthEast::ragelParseCoordinatesNe (std::string *data*) [private]**

Populate mCoordinates.

Definition at line 93 of file sosi\_north\_east\_ragel.cpp.

**void sosicon::sosi::SosiNorthEast::ragelParseCoordinatesNeh (std::string *data*) [private]**

Definition at line 100 of file sosi\_north\_east\_height\_ragel.cpp.

**void sosicon::sosi::SosiNorthEast::reverse () [inline]**

Reverse polygon (point order)

Definition at line 105 of file sosi\_north\_east.h.

---

## Member Data Documentation

**CoordinateList sosicon::sosi::SosiNorthEast::mCoordinates [private]**

Definition at line 54 of file sosi\_north\_east.h.

**CoordinateList::iterator sosicon::sosi::SosiNorthEast::mCoordinatesIterator [private]**

Definition at line 56 of file sosi\_north\_east.h.

**double sosicon::sosi::SosiNorthEast::mMaxX [private]**

Definition at line 64 of file sosi\_north\_east.h.

**double sosicon::sosi::SosiNorthEast::mMaxY [private]**

Definition at line 65 of file sosi\_north\_east.h.

**double sosicon::sosi::SosiNorthEast::mMinX [private]**

Definition at line 62 of file sosi\_north\_east.h.

**double sosicon::sosi::SosiNorthEast::mMinY [private]**

Definition at line 63 of file sosi\_north\_east.h.

**sosicon::sosi::SosiOrigoNE sosicon::sosi::SosiNorthEast::mOrigo [static], [private]**

Definition at line 58 of file sosi\_north\_east.h.

**ISosiElement\* sosicon::sosi::SosiNorthEast::mSosiElement [private]**

Definition at line 52 of file sosi\_north\_east.h.

**sosicon::sosi::SosiUnit sosicon::sosi::SosiNorthEast::mUnit [static], [private]**

Definition at line 60 of file sosi\_north\_east.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/**sosi\_north\_east.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/**sosi\_north\_east.cpp**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**sosi\_north\_east\_height\_ragel.cpp**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**sosi\_north\_east\_ragel.cpp**

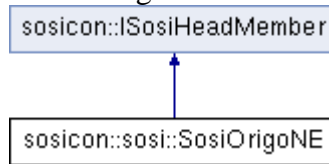


## sosicon::sosi::SosiOrigoNE Class Reference

SOSI Junction point.

```
#include <sosi_origo_ne.h>
```

Inheritance diagram for sosicon::sosi::SosiOrigoNE:



### Public Member Functions

- **SosiOrigoNE ()**  
*Construct new SOSI origo element.*
- **SosiOrigoNE (ISosiElement \*e)**  
*Construct new SOSI junction point element.*
- **virtual ~SosiOrigoNE ()**  
*Destructor.*
- **int getN ()**
- **int getE ()**
- **virtual void init (ISosiElement \*e)**
- **virtual bool initialized ()**
- **void ragelParseSosiOrigoNE (std::string data)**  
*Ragel parse element data.*

### Private Attributes

- **ISosiElement \* mSosiElement**
- **bool mInitialized**
- **int mOrigoN**
- **int mOrigoE**

---

### Detailed Description

SOSI Junction point.

Implements SOSI junction point layer specification, as given via the KP element. Norwegian: Knutepunktslag.

Definition at line 44 of file `sosi_origo_ne.h`.

---

### Constructor & Destructor Documentation

#### sosicon::sosi::SosiOrigoNE::SosiOrigoNE ()

Construct new SOSI origo element.

Definition at line 21 of file `sosi_origo_ne.cpp`.

**sosicon::sosi::SosiOrigoNE::SosiOrigoNE (ISosiElement \* e)[inline]**

Construct new SOSI junction point element.

Definition at line 60 of file sosi\_origo\_ne.h.

**virtual sosicon::sosi::SosiOrigoNE::~~SosiOrigoNE () [inline], [virtual]**

Destructor.

Definition at line 63 of file sosi\_origo\_ne.h.

---

## Member Function Documentation

**int sosicon::sosi::SosiOrigoNE::getE () [inline]**

Definition at line 67 of file sosi\_origo\_ne.h.

**int sosicon::sosi::SosiOrigoNE::getN () [inline]**

Definition at line 65 of file sosi\_origo\_ne.h.

**void sosicon::sosi::SosiOrigoNE::init (ISosiElement \* e) [virtual]**

Implements **sosicon::ISosiHeadMember** (*p.116*).

Definition at line 28 of file sosi\_origo\_ne.cpp.

**virtual bool sosicon::sosi::SosiOrigoNE::initialized () [inline], [virtual]**

Implements **sosicon::ISosiHeadMember** (*p.116*).

Definition at line 71 of file sosi\_origo\_ne.h.

**void sosicon::sosi::SosiOrigoNE::ragelParseSosiOrigoNE (std::string data)**

Ragel parse element data.

Definition at line 88 of file sosi\_origo\_ne\_ragel.cpp.

---

## Member Data Documentation

**bool sosicon::sosi::SosiOrigoNE::mInitialized [private]**

Definition at line 48 of file sosi\_origo\_ne.h.

**int sosicon::sosi::SosiOrigoNE::mOrigoE [private]**

Definition at line 52 of file sosi\_origo\_ne.h.

**int sosicon::sosi::SosiOrigoNE::mOrigoN [private]**

Definition at line 50 of file sosi\_origo\_ne.h.

**ISosiElement\* sosicon::sosi::SosiOrigoNE::mSosiElement [private]**

Definition at line 46 of file sosi\_origo\_ne.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_origo\_ne.h
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_origo\_ne.cpp
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi\_origo\_ne\_ragel.cpp

## sosicon::sosi::SosiRefList Class Reference

SOSI REF list.

```
#include <sosi_ref_list.h>
```

### Public Member Functions

- **SosiRefList ()**  
*Construct new SOSI REF element.*
- **SosiRefList (ISosiElement \*e)**  
*Construct new SOSI REF element.*
- **~SosiRefList ()**  
*Destructor.*
- **bool getNextGeometry (GeometryRef \*&geometry)**  
*Next list of references.*

### Private Member Functions

- **void ragelParseSosiRef (std::string data)**

### Private Attributes

- **ISosiElement \* mSosiElement**
- **GeometryCollection mRefListCollection**
- **GeometryCollection::size\_type mRefListCollectionIndex**
- **GeometryRef::size\_type mRefListIndex**

---

## Detailed Description

SOSI REF list.

Implements SOSI reference list, as given via the REF element.

Definition at line 42 of file `sosi_ref_list.h`.

---

## Constructor & Destructor Documentation

### sosicon::sosi::SosiRefList::SosiRefList () [inline]

Construct new SOSI REF element.

Definition at line 56 of file `sosi_ref_list.h`.

### sosicon::sosi::SosiRefList::SosiRefList (ISosiElement \* e)

Construct new SOSI REF element.

Definition at line 21 of file `sosi_ref_list.cpp`.

## **sosicon::sosi::SosiRefList::~~SosiRefList ()**

Destructor.

Definition at line 28 of file sosi\_ref\_list.cpp.

---

## **Member Function Documentation**

### **bool sosicon::sosi::SosiRefList::getNextGeometry (GeometryRef \*& geometry)**

Next list of references.

A reference list represents a geometry, i.e. a polygon or its holes.

Definition at line 40 of file sosi\_ref\_list.cpp.

### **void sosicon::sosi::SosiRefList::ragelParseSosiRef (std::string data) [private]**

Definition at line 90 of file sosi\_ref\_ragel.cpp.

---

## **Member Data Documentation**

### **GeometryCollection sosicon::sosi::SosiRefList::mRefListCollection [private]**

Definition at line 46 of file sosi\_ref\_list.h.

### **GeometryCollection::size\_type sosicon::sosi::SosiRefList::mRefListCollectionIndex [private]**

Definition at line 48 of file sosi\_ref\_list.h.

### **GeometryRef::size\_type sosicon::sosi::SosiRefList::mRefListIndex [private]**

Definition at line 49 of file sosi\_ref\_list.h.

### **ISosiElement\* sosicon::sosi::SosiRefList::mSosiElement [private]**

Definition at line 44 of file sosi\_ref\_list.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/**sosi\_ref\_list.h**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/**sosi\_ref\_list.cpp**
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/**sosi\_ref\_ragel.cpp**

## sosicon::sosi::SosiTranslationTable Class Reference

```
#include <sosi_translation_table.h>
```

### Public Member Functions

- **SosiTranslationTable** ()
- **CoordSys & sysCodeToCoordSys** (int sysCode)
- **ElementType sosiNameToType** (std::string typeName)
- **std::string sosiTypeToName** (**ElementType** elementType)
- **ObjType sosiObjNameToType** (std::string objTypeName)
- **std::string sosiTypeToObjName** (**ObjType** objType)

### Private Member Functions

- **template<typename Key , typename Val > Key reverseLookup** (std::map< Key, Val > &c, Val v)  
*Scan container looking for value, returning key.*

### Private Attributes

- **SosiCharsetSingleton \* mSosiCharset**  
*Character encoding element.*

### Static Private Attributes

- **static const int MAX\_COORDSYS\_TABLE**  
*Number of entries in KOORDSYS lookup table.*
- **static std::map< std::string, ElementType > mTypeNameMap**  
*SOSI element name map.*
- **static std::map< std::string, ObjType > mObjTypeNameMap**  
*OBJTYPE name lookup table.*
- **static CoordSys mCoordSysTable [MAX\_COORDSYS\_TABLE+1]**  
*KOORDSYS code lookup table.*

---

## Detailed Description

Definition at line 37 of file sosi\_translation\_table.h.

---

## Constructor & Destructor Documentation

### sosicon::sosi::SosiTranslationTable::SosiTranslationTable ()

Definition at line 26 of file sosi\_translation\_table.cpp.

---

## Member Function Documentation

**template<typename Key , typename Val > Key  
sosicon::sosi::SosiTranslationTable::reverseLookup (std::map< Key, Val > & c, Val v)[inline],  
[private]**

Scan container looking for value, returning key.

Definition at line 67 of file sosi\_translation\_table.h.

**ElementType sosicon::sosi::SosiTranslationTable::sosiNameToType (std::string  
typeName)[inline]**

Definition at line 83 of file sosi\_translation\_table.h.

**ObjType sosicon::sosi::SosiTranslationTable::sosiObjNameToType (std::string  
objTypeName)[inline]**

Definition at line 92 of file sosi\_translation\_table.h.

**std::string sosicon::sosi::SosiTranslationTable::sosiTypeToName (ElementType  
elementType)[inline]**

Definition at line 88 of file sosi\_translation\_table.h.

**std::string sosicon::sosi::SosiTranslationTable::sosiTypeToObjName (ObjType  
objType)[inline]**

Definition at line 97 of file sosi\_translation\_table.h.

**CoordSys& sosicon::sosi::SosiTranslationTable::sysCodeToCoordSys (int sysCode)[inline]**

Definition at line 78 of file sosi\_translation\_table.h.

---

## Member Data Documentation

**const int sosicon::sosi::SosiTranslationTable::MAX\_COORDSYS\_TABLE [static], [private]**

Number of entries in KOORDSYS lookup table.

Definition at line 43 of file sosi\_translation\_table.h.

**sosicon::sosi::CoordSys sosicon::sosi::SosiTranslationTable::mCoordSysTable [static],  
[private]**

KOORDSYS code lookup table.

Primitive array, int key.

Definition at line 63 of file `sosi_translation_table.h`.

```
std::map< std::string, sosicon::sosi::ObjType >  
sosicon::sosi::SosiTranslationTable::mObjTypeNameMap [static], [private]
```

OBJTYPE name lookup table.

STL map with string key.

Definition at line 57 of file `sosi_translation_table.h`.

```
SosiCharsetSingleton* sosicon::sosi::SosiTranslationTable::mSosiCharset [private]
```

Character encoding element.

Definition at line 40 of file `sosi_translation_table.h`.

```
std::map< std::string, sosicon::sosi::ElementType >  
sosicon::sosi::SosiTranslationTable::mTypeNameMap [static], [private]
```

SOSI element name map.

STL map with string key.

Definition at line 51 of file `sosi_translation_table.h`.

---

**The documentation for this class was generated from the following files:**

- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_translation_table.h`
- `/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi_translation_table.cpp`



## sosicon::sosi::SosiUnit Class Reference

SOSI Unit.

```
#include <sosi_unit.h>
```

Inheritance diagram for sosicon::sosi::SosiUnit:



### Public Member Functions

- **SosiUnit ()**  
*Construct new SOSI Unit element.*
- **virtual ~SosiUnit ()**  
*Destructor.*
- **SosiUnit (ISosiElement \*e)**  
*Construct new SOSI Unit element.*
- **int getDivisor ()**
- **virtual void init (ISosiElement \*e)**  
*Initialize SOSI Unit element.*
- **virtual bool initialized ()**

### Private Attributes

- **ISosiElement \* mSosiElement**
- **bool mInitialized**
- **int mDivisor**

---

## Detailed Description

SOSI Unit.

Implements SOSI unit (coordinate resolution), as given via the ENHET element.

Definition at line 44 of file `sosi_unit.h`.

---

## Constructor & Destructor Documentation

### sosicon::sosi::SosiUnit::SosiUnit ()

Construct new SOSI Unit element.

Definition at line 21 of file `sosi_unit.cpp`.

### virtual sosicon::sosi::SosiUnit::~~SosiUnit () [inline], [virtual]

Destructor.

Definition at line 58 of file sosi\_unit.h.

**sosicon::sosi::SosiUnit::SosiUnit (ISosiElement \* e)[inline]**

Construct new SOSI Unit element.

Definition at line 61 of file sosi\_unit.h.

---

## Member Function Documentation

**int sosicon::sosi::SosiUnit::getDivisor () [inline]**

Definition at line 63 of file sosi\_unit.h.

**void sosicon::sosi::SosiUnit::init (ISosiElement \* e)[virtual]**

Initnialize SOSI Unit element.

Implements **sosicon::ISosiHeadMember** (*p.116*).

Definition at line 28 of file sosi\_unit.cpp.

**virtual bool sosicon::sosi::SosiUnit::initialized () [inline], [virtual]**

Implements **sosicon::ISosiHeadMember** (*p.116*).

Definition at line 68 of file sosi\_unit.h.

---

## Member Data Documentation

**int sosicon::sosi::SosiUnit::mDivisor [private]**

Definition at line 50 of file sosi\_unit.h.

**bool sosicon::sosi::SosiUnit::mIntialized [private]**

Definition at line 48 of file sosi\_unit.h.

**ISosiElement\* sosicon::sosi::SosiUnit::mSosiElement [private]**

Definition at line 46 of file sosi\_unit.h.

---

**The documentation for this class was generated from the following files:**

- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_unit.h
- /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_unit.cpp

# File Documentation

## /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/byte\_order.cpp File Reference

```
#include "byte_order.h"
```

### Variables

- `enum sosicon::byteOrder::Endianness sosicon`

---

### Variable Documentation

`enum sosicon::byteOrder::Endianness sosicon`

Definition at line 21 of file `byte_order.cpp`.

## /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/byte\_order.h

### File Reference

```
#include <inttypes.h>
#include "logger.h"
#include <algorithm>
#include <cmath>
```

### Namespaces

- **sosicon**
- *Application root.* **sosicon::byteOrder**

### Big/low-endian conversions. Enumerations

- enum **sosicon::byteOrder::Endianness** { **sosicon::byteOrder::not\_set**, **sosicon::byteOrder::big**, **sosicon::byteOrder::little** } *Big/little flag.*

### Functions

- Endianness **sosicon::byteOrder::determine** ()  
*Determines system endianness.*
- void **sosicon::byteOrder::doubleToLittleEndian** (double from, char \*to)  
*Writes little endian representation of double.*
- void **sosicon::byteOrder::toBigEndian** (const char \*from, char \*to, size\_t bufSize)  
*Reverses buffer to big endian if required.*
- void **sosicon::byteOrder::toLittleEndian** (const char \*from, char \*to, size\_t bufSize)  
*Reverses buffer to little endian if required.*

### Variables

- enum **sosicon::byteOrder::Endianness**                      **sosicon::byteOrder::endianness**  
*Stores system endianness.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/command\_line.cpp File Reference**

```
#include "command_line.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/command\_line.h File Reference**

```
#include <stdio.h>
#include <iostream>
#include <vector>
#include <string>
#include "logger.h"
#include <unistd.h>
#include "utils.h"
```

### **Classes**

- class **sosicon::CommandLine**

### ***Command-line parser.* Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/common\_types.h File Reference**

```
#include <vector>
#include "interface/i_coordinate.h"
```

### **Namespaces**

- **sosicon**

### ***Application root. Typedefs***

- `typedef std::vector< ICoordinate * > sosicon::CoordinateList`  
*List of coordinate pairs.*

### **Enumerations**

- `enum sosicon::Wkt { sosicon::wkt_unknown, sosicon::wkt_point, sosicon::wkt_linestring, sosicon::wkt_polygon }` *List of applied, well-known text geometries.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter\_sosi2psql.cpp File Reference**

```
#include "converter_sosi2psql.h"
```



## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter\_sosi2psql.h File Reference**

```
#include "logger.h"
#include <fstream>
#include <sstream>
#include <vector>
#include <climits>
#include <cmath>
#include <map>
#include "utils.h"
#include "interface/i_converter.h"
#include "interface/i_sosi_element.h"
#include "sosi/sosi_types.h"
#include "sosi/sosi_translation_table.h"
#include "coordinate_collection.h"
#include "sosi/sosi_north_east.h"
#include "command_line.h"
#include "common_types.h"
#include "parser.h"
```

### **Classes**

- class **sosicon::ConverterSosi2psql**
- *SOSI to PostgreSQL/PostGIS converter.* class **sosicon::ConverterSosi2psql::Field**

### **Namespaces**

- **sosicon**  
*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter\_sosi2shp.cpp File Reference**

```
#include "converter_sosi2shp.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter\_sosi2shp.h File Reference**

```
#include "logger.h"
#include <iomanip>
#include <fstream>
#include <vector>
#include <sstream>
#include <string>
#include "interface/i_converter.h"
#include "interface/i_sosi_element.h"
#include "command_line.h"
#include "parser.h"
#include "utils.h"
#include "shape/shapefile.h"
#include <sys/stat.h>
#include <sys/types.h>
```

### **Classes**

- class **sosicon::ConverterSosi2shp**

### ***SOSI to ESRI Shape converter.* Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter\_sosi2tsv.cpp File Reference**

```
#include "converter_sosi2tsv.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter\_sosi2tsv.h File Reference**

```
#include <iostream>
#include <fstream>
#include <vector>
#include "interface/i_converter.h"
#include "command_line.h"
#include "parser.h"
```

### **Classes**

- class **sosicon::ConverterSosi2tsv**

### ***SOSI to TSV converter.* Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter\_sosi2xml.cpp File Reference**

```
#include "converter_sosi2xml.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter\_sosi2xml.h File Reference**

```
#include <iostream>
#include <fstream>
#include <vector>
#include "interface/i_converter.h"
#include "interface/i_sosi_element.h"
#include "command_line.h"
#include "utils.h"
#include "parser.h"
```

### **Classes**

- class **sosicon::ConverterSosi2xml**

### ***SOSI to ESRI Shape converter.* Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter\_sosi\_stat.cpp File Reference**

```
#include "converter_sosi_stat.h"
```



## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/converter\_sosi\_stat.h File Reference**

```
#include "logger.h"
#include <fstream>
#include <map>
#include "interface/i_converter.h"
#include "interface/i_sosi_element.h"
#include "sosi/sosi_types.h"
#include "sosi/sosi_element_search.h"
#include "command_line.h"
#include "utils.h"
#include "parser.h"
```

### **Classes**

- class **sosicon::ConverterSosiStat**

### ***SOSI to ESRI Shape converter.* Namespaces**

- **sosicon**

*Application root.*

**/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate.h**

## **File Reference**

```
#include <string>
#include <iostream>
#include <sstream>
#include <ios>
#include "interface/i_coordinate.h"
```

## **Classes**

- class **sosicon::Coordinate**

## *Coordinate* **container**. Namespaces

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate\_collection.cpp File Reference**

```
#include "coordinate_collection.h"
```

## /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/coordinate\_collection.h File Reference

```
#include <algorithm>
#include <limits>
#include <vector>
#include "logger.h"
#include "common_types.h"
#include "sosi/sosi_types.h"
#include "sosi/sosi_element_search.h"
#include "sosi/sosi_ref_list.h"
#include "sosi/sosi_north_east.h"
#include "interface/i_coordinate.h"
#include "interface/i_sosi_element.h"
```

### Classes

- class **sosicon::CoordinateCollection**

### Coordinate *container*. Namespaces

- **sosicon**

### Application root. Functions

- bool **sosicon::getNext** (ICoordinate \*&coord, sosi::NorthEastList &list, sosi::NorthEastList::iterator &i)  
*Get next coordinate in list.*
- bool **sosicon::getNextOffset** (int &offset, std::vector< int > &offsets, std::vector< int >::iterator &iterator)  
*Get next offset in part offsets list.*
- bool **sosicon::isClockwise** (std::vector< ICoordinate \* >::iterator &begin, std::vector< ICoordinate \* >::iterator &end)  
*Analyzes polygon direction.*
- bool **sosicon::isCounterClockwise** (std::vector< ICoordinate \* >::iterator &begin, std::vector< ICoordinate \* >::iterator &end)  
*Analyzes polygon direction.*
- void **sosicon::neListToCoordList** (sosi::NorthEastList &neList, std::vector< ICoordinate \* > &coordList)  
*Extracts single coordinates from list of North-East elements.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/event\_dispatcher.h File Reference**

```
#include <memory>
#include <vector>
#include <iostream>
```

### **Classes**

- class **sosicon::EventDispatcher< Event >**
- *Event dispatcher template class.* class **sosicon::EventDispatcher< Event >::Listener**

### **Namespaces**

- **sosicon**  
*Application root.*

**/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/factory.cpp**

## **File Reference**

```
#include "factory.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/factory.h File Reference**

```
#include "interface/i_converter.h"
#include "converter_sosi2shp.h"
#include "converter_sosi2xml.h"
#include "converter_sosi2tsv.h"
#include "converter_sosi2psql.h"
#include "converter_sosi_stat.h"
```

### **Classes**

- class **sosicon::Factory**

### ***Factory class. Namespaces***

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_binary\_streamable.h File Reference**

`#include <iostream>`

### **Classes**

- class `sosicon::IBinaryStreamable`

### ***Interface: Binary streamable object.* Namespaces**

- `sosicon`

### ***Application root.* Functions**

- `std::ostream & sosicon::operator<< (std::ostream &os, IBinaryStreamable &binaryStreamable)`  
*Stream output operator.*



## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_converter.h File Reference**

```
#include "../command_line.h"
```

### **Classes**

- class **sosicon::IConverter**

### ***Interface: Converter.* Namespaces**

- **sosicon**

*Application root.*

## /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_coordinate.h File Reference

```
#include <string>
```

### Classes

- class `sosicon::ICoordinate`

### ***Interface:*** *Coordinate*. Namespaces

- `sosicon`

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_lookup\_table.h File Reference**

```
#include <string>
```

### **Classes**

- class `sosicon::ILookupTable`

### ***Interface: Lookup table.* Namespaces**

- `sosicon`

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_rectangle.h File Reference**

### **Classes**

- `class sosicon::IRectangle`

### ***Interface: Rectangle.* Namespaces**

- `sosicon`

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shape\_element.h File Reference**

```
#include "i_binary_streamable.h"
#include "i_sosi_element.h"
#include "i_rectangle.h"
```

### **Classes**

- class **sosicon::IShapeElement**

### ***Interface: Shape element.* Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shape\_element\_header.h File Reference**

```
#include "i_binary_streamable.h"
```

### **Classes**

- class `sosicon::IShapeElementHeader`

### ***Interface: Shape element header.* Namespaces**

- `sosicon`

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shape\_header.h File Reference**

```
#include "i_binary_streamable.h"
#include "../shape/shapefile_types.h"
#include "../shape/bounding_box.h"
```

### **Classes**

- class **sosicon::IShapeHeader**

### ***Interface: Shape element.* Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shapefile.h File Reference**

```
#include "i_shapefile_shp_part.h"
#include "i_shapefile_shx_part.h"
#include "i_shapefile_dbf_part.h"
#include "i_shapefile_prj_part.h"
#include "i_sosi_element.h"
#include "../sosi/sosi_types.h"
```

### **Classes**

- class **sosicon::IShapefile**

### ***Interface: Shapefile.* Namespaces**

- **sosicon**

*Application root.*



## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shapefile\_dbf\_part.h File Reference**

```
#include <iostream>
#include "i_binary_streamable.h"
```

### **Classes**

- class **sosicon::IShapefileDbfPart**

### **Interface: ShapefileDbfPart. Namespaces**

- **sosicon**  
*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shapefile\_prj\_part.h File Reference**

```
#include <iostream>
#include "i_binary_streamable.h"
```

### **Classes**

- class **sosicon::IShapefilePrjPart**

### **Interface: ShapefilePrjPart. Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shapefile\_shp\_part.h File Reference**

```
#include <iostream>
#include "i_binary_streamable.h"
```

### **Classes**

- class **sosicon::IShapefileShpPart**

### **Interface: ShapefileShpPart. Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_shapefile\_shx\_part.h File Reference**

```
#include <iostream>
#include "i_binary_streamable.h"
```

### **Classes**

- class **sosicon::IShapefileShxPart**

### ***Interface: ShapefileShxPart.* Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_sosi\_element.h File Reference**

```
#include <string>
#include <vector>
#include <map>
#include "../sosi/sosi_types.h"
#include "../sosi/sosi_element_search.h"
```

### **Classes**

- class **sosicon::ISosiElement**

### ***Interface: SOSI element.* Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/interface/i\_sosi\_head\_member.h File Reference**

```
#include "i_sosi_element.h"
```

### **Classes**

- class `sosicon::ISosiHeadMember`

### ***Interface: SOSI header element.* Namespaces**

- `sosicon`

*Application root.*

```
#include "stdint.h"
```

- struct **imaxdiv\_t**

```
• #define PRId8
• #define PRIi8
• #define PRIdLEAST8
• #define PRIiLEAST8
• #define PRIdFAST8
• #define PRIiFAST8
• #define PRId16
• #define PRIi16
• #define PRIdLEAST16
• #define PRIiLEAST16
• #define PRIdFAST16
• #define PRIiFAST16
• #define PRId32
• #define PRIi32
• #define PRIdLEAST32
• #define PRIiLEAST32
• #define PRIdFAST32
• #define PRIiFAST32
• #define PRId64
• #define PRIi64
• #define PRIdLEAST64
• #define PRIiLEAST64
• #define PRIdFAST64
• #define PRIiFAST64
• #define PRIdMAX
• #define PRIiMAX
• #define PRIdPTR
• #define PRIiPTR
• #define PRIO8
• #define PRIu8
• #define PRIx8
• #define PRIX8
• #define PRIOLEAST8
• #define PRIULEAST8
• #define PRIXLEAST8
• #define PRILEAST8
• #define PRIOFAST8
• #define PRIUFAST8
• #define PRIXFAST8
• #define PRIFAST8
• #define PRIO16
• #define PRIu16
```

- **#define PRIx16**
- **#define PRIX16**
- **#define PRIoLEAST16**
- **#define PRIuLEAST16**
- **#define PRIxLEAST16**
- **#define PRIXLEAST16**
- **#define PRIoFAST16**
- **#define PRIuFAST16**
- **#define PRIxFAST16**
- **#define PRIXFAST16**
- **#define PRIo32**
- **#define PRIu32**
- **#define PRIx32**
- **#define PRIX32**
- **#define PRIoLEAST32**
- **#define PRIuLEAST32**
- **#define PRIxLEAST32**
- **#define PRIXLEAST32**
- **#define PRIoFAST32**
- **#define PRIuFAST32**
- **#define PRIxFAST32**
- **#define PRIXFAST32**
- **#define PRIo64**
- **#define PRIu64**
- **#define PRIx64**
- **#define PRIX64**
- **#define PRIoLEAST64**
- **#define PRIuLEAST64**
- **#define PRIxLEAST64**
- **#define PRIXLEAST64**
- **#define PRIoFAST64**
- **#define PRIuFAST64**
- **#define PRIxFAST64**
- **#define PRIXFAST64**
- **#define PRIoMAX**
- **#define PRIuMAX**
- **#define PRIxMAX**
- **#define PRIXMAX**
- **#define PRIoPTR**
- **#define PRIuPTR**
- **#define PRIxPTR**
- **#define PRIPTR**
- **#define SCNd8**
- **#define SCNi8**
- **#define SCNdLEAST8**
- **#define SCNiLEAST8**
- **#define SCNdFAST8**
- **#define SCNiFAST8**
- **#define SCNd16**
- **#define SCNi16**
- **#define SCNdLEAST16**
- **#define SCNiLEAST16**
- **#define SCNdFAST16**



- **#define SCNiFAST16**
- **#define SCNd32**
- **#define SCNi32**
- **#define SCNdLEAST32**
- **#define SCNiLEAST32**
- **#define SCNdFAST32**
- **#define SCNiFAST32**
- **#define SCNd64**
- **#define SCNi64**
- **#define SCNdLEAST64**
- **#define SCNiLEAST64**
- **#define SCNdFAST64**
- **#define SCNiFAST64**
- **#define SCNdMAX**
- **#define SCNiMAX**
- **#define SCNdPTR**
- **#define SCNiPTR**
- **#define SCNo8**
- **#define SCNu8**
- **#define SCNx8**
- **#define SCNX8**
- **#define SCNoLEAST8**
- **#define SCNuLEAST8**
- **#define SCNxLEAST8**
- **#define SCNXLEAST8**
- **#define SCNoFAST8**
- **#define SCNuFAST8**
- **#define SCNxFAST8**
- **#define SCNXFAST8**
- **#define SCNo16**
- **#define SCNu16**
- **#define SCNx16**
- **#define SCNX16**
- **#define SCNoLEAST16**
- **#define SCNuLEAST16**
- **#define SCNxLEAST16**
- **#define SCNXLEAST16**
- **#define SCNoFAST16**
- **#define SCNuFAST16**
- **#define SCNxFAST16**
- **#define SCNXFAST16**
- **#define SCNo32**
- **#define SCNu32**
- **#define SCNx32**
- **#define SCNX32**
- **#define SCNoLEAST32**
- **#define SCNuLEAST32**
- **#define SCNxLEAST32**
- **#define SCNXLEAST32**
- **#define SCNoFAST32**
- **#define SCNuFAST32**
- **#define SCNxFAST32**
- **#define SCNXFAST32**

- **#define SCNo64**
- **#define SCNu64**
- **#define SCNx64**
- **#define SCNX64**
- **#define SCNoLEAST64**
- **#define SCNuLEAST64**
- **#define SCNxLEAST64**
- **#define SCNXLEAST64**
- **#define SCNoFAST64**
- **#define SCNuFAST64**
- **#define SCNxFAST64**
- **#define SCNXFAST64**
- **#define SCNoMAX**
- **#define SCNuMAX**
- **#define SCNxMAX**
- **#define SCNXMAX**
- **#define SCNoPTR**
- **#define SCNuPTR**
- **#define SCNxPTR**
- **#define SCNXPTR**
- **#define imaxabs**
- **#define strtoumax**
- **#define wcstoumax**

## Functions

- **imaxdiv\_t \_\_cdecl imaxdiv** (intmax\_t numer, intmax\_t denom)

---

## Macro Definition Documentation

### **#define imaxabs**

Definition at line 269 of file inttypes.h.

### **#define PRId16**

Definition at line 64 of file inttypes.h.

### **#define PRId32**

Definition at line 71 of file inttypes.h.

### **#define PRId64**

Definition at line 78 of file inttypes.h.

**#define PRId8**

Definition at line 57 of file inttypes.h.

**#define PRIdFAST16**

Definition at line 68 of file inttypes.h.

**#define PRIdFAST32**

Definition at line 75 of file inttypes.h.

**#define PRIdFAST64**

Definition at line 82 of file inttypes.h.

**#define PRIdFAST8**

Definition at line 61 of file inttypes.h.

**#define PRIdLEAST16**

Definition at line 66 of file inttypes.h.

**#define PRIdLEAST32**

Definition at line 73 of file inttypes.h.

**#define PRIdLEAST64**

Definition at line 80 of file inttypes.h.

**#define PRIdLEAST8**

Definition at line 59 of file inttypes.h.

**#define PRIdMAX**

Definition at line 85 of file inttypes.h.

**#define PRIdPTR**

Definition at line 88 of file inttypes.h.

**#define PRIi16**

Definition at line 65 of file inttypes.h.

**#define PRIi32**

Definition at line 72 of file inttypes.h.

**#define PRIi64**

Definition at line 79 of file inttypes.h.

**#define PRIi8**

Definition at line 58 of file inttypes.h.

**#define PRIiFAST16**

Definition at line 69 of file inttypes.h.

**#define PRIiFAST32**

Definition at line 76 of file inttypes.h.

**#define PRIiFAST64**

Definition at line 83 of file inttypes.h.

**#define PRIiFAST8**

Definition at line 62 of file inttypes.h.

**#define PRIiLEAST16**

Definition at line 67 of file inttypes.h.

**#define PRIiLEAST32**

Definition at line 74 of file inttypes.h.

**#define PRIiLEAST64**

Definition at line 81 of file inttypes.h.

**#define PRIiLEAST8**

Definition at line 60 of file inttypes.h.

**#define PRIiMAX**

Definition at line 86 of file inttypes.h.

**#define PRIiPTR**

Definition at line 89 of file inttypes.h.

**#define PRIo16**

Definition at line 105 of file inttypes.h.

**#define PRIo32**

Definition at line 118 of file inttypes.h.

**#define PRIo64**

Definition at line 131 of file inttypes.h.

**#define PRIo8**

Definition at line 92 of file inttypes.h.

**#define PRIoFAST16**

Definition at line 113 of file inttypes.h.

**#define PRIoFAST32**

Definition at line 126 of file inttypes.h.

**#define PRIoFAST64**

Definition at line 139 of file inttypes.h.

**#define PRIoFAST8**

Definition at line 100 of file inttypes.h.

**#define PRIoLEAST16**

Definition at line 109 of file inttypes.h.

**#define PRIoLEAST32**

Definition at line 122 of file inttypes.h.

**#define PRIoLEAST64**

Definition at line 135 of file inttypes.h.

**#define PRIoLEAST8**

Definition at line 96 of file inttypes.h.

**#define PRIoMAX**

Definition at line 144 of file inttypes.h.

**#define PRIoPTR**

Definition at line 149 of file inttypes.h.

**#define PRIu16**

Definition at line 106 of file inttypes.h.

**#define PRIu32**

Definition at line 119 of file inttypes.h.

**#define PRIu64**

Definition at line 132 of file inttypes.h.

**#define PRIu8**

Definition at line 93 of file inttypes.h.

**#define PRIuFAST16**

Definition at line 114 of file inttypes.h.

**#define PRIuFAST32**

Definition at line 127 of file inttypes.h.

**#define PRIuFAST64**

Definition at line 140 of file inttypes.h.

**#define PRIuFAST8**

Definition at line 101 of file inttypes.h.

**#define PRIuLEAST16**

Definition at line 110 of file inttypes.h.

**#define PRIuLEAST32**

Definition at line 123 of file inttypes.h.

**#define PRIuLEAST64**

Definition at line 136 of file inttypes.h.

**#define PRIuLEAST8**

Definition at line 97 of file inttypes.h.

**#define PRIuMAX**

Definition at line 145 of file inttypes.h.

**#define PRIuPTR**

Definition at line 150 of file inttypes.h.

**#define PRIx16**

Definition at line 107 of file inttypes.h.

**#define PRIx16**

Definition at line 108 of file inttypes.h.

**#define PRlx32**

Definition at line 120 of file inttypes.h.

**#define PRlX32**

Definition at line 121 of file inttypes.h.

**#define PRlx64**

Definition at line 133 of file inttypes.h.

**#define PRlX64**

Definition at line 134 of file inttypes.h.

**#define PRlx8**

Definition at line 94 of file inttypes.h.

**#define PRlX8**

Definition at line 95 of file inttypes.h.

**#define PRlxFast16**

Definition at line 115 of file inttypes.h.

**#define PRlXFast16**

Definition at line 116 of file inttypes.h.

**#define PRlxFast32**

Definition at line 128 of file inttypes.h.

**#define PRlXFast32**

Definition at line 129 of file inttypes.h.

**#define PRlxFast64**

Definition at line 141 of file inttypes.h.



**#define PRIFAST64**

Definition at line 142 of file inttypes.h.

**#define PRIfAST8**

Definition at line 102 of file inttypes.h.

**#define PRIFAST8**

Definition at line 103 of file inttypes.h.

**#define PRIfLEAST16**

Definition at line 111 of file inttypes.h.

**#define PRILEAST16**

Definition at line 112 of file inttypes.h.

**#define PRIfLEAST32**

Definition at line 124 of file inttypes.h.

**#define PRILEAST32**

Definition at line 125 of file inttypes.h.

**#define PRIfLEAST64**

Definition at line 137 of file inttypes.h.

**#define PRILEAST64**

Definition at line 138 of file inttypes.h.

**#define PRIfLEAST8**

Definition at line 98 of file inttypes.h.

**#define PRILEAST8**

Definition at line 99 of file inttypes.h.

**#define PRIxMAX**

Definition at line 146 of file inttypes.h.

**#define PRIXMAX**

Definition at line 147 of file inttypes.h.

**#define PRIxPTR**

Definition at line 151 of file inttypes.h.

**#define PRIXPTR**

Definition at line 152 of file inttypes.h.

**#define SCNd16**

Definition at line 162 of file inttypes.h.

**#define SCNd32**

Definition at line 169 of file inttypes.h.

**#define SCNd64**

Definition at line 176 of file inttypes.h.

**#define SCNd8**

Definition at line 155 of file inttypes.h.

**#define SCNdFAST16**

Definition at line 166 of file inttypes.h.

**#define SCNdFAST32**

Definition at line 173 of file inttypes.h.

**#define SCNdFAST64**

Definition at line 180 of file inttypes.h.

**#define SCNdFAST8**

Definition at line 159 of file inttypes.h.

**#define SCNdLEAST16**

Definition at line 164 of file inttypes.h.

**#define SCNdLEAST32**

Definition at line 171 of file inttypes.h.

**#define SCNdLEAST64**

Definition at line 178 of file inttypes.h.

**#define SCNdLEAST8**

Definition at line 157 of file inttypes.h.

**#define SCNdMAX**

Definition at line 183 of file inttypes.h.

**#define SCNdPTR**

Definition at line 190 of file inttypes.h.

**#define SCNi16**

Definition at line 163 of file inttypes.h.

**#define SCNi32**

Definition at line 170 of file inttypes.h.

**#define SCNi64**

Definition at line 177 of file inttypes.h.

**#define SCNi8**

Definition at line 156 of file inttypes.h.

**#define SCNiFAST16**

Definition at line 167 of file inttypes.h.

**#define SCNiFAST32**

Definition at line 174 of file inttypes.h.

**#define SCNiFAST64**

Definition at line 181 of file inttypes.h.

**#define SCNiFAST8**

Definition at line 160 of file inttypes.h.

**#define SCNiLEAST16**

Definition at line 165 of file inttypes.h.

**#define SCNiLEAST32**

Definition at line 172 of file inttypes.h.

**#define SCNiLEAST64**

Definition at line 179 of file inttypes.h.

**#define SCNiLEAST8**

Definition at line 158 of file inttypes.h.

**#define SCNiMAX**

Definition at line 184 of file inttypes.h.

**#define SCNiPTR**

Definition at line 191 of file inttypes.h.

**#define SCNo16**

Definition at line 208 of file inttypes.h.

**#define SCNo32**

Definition at line 221 of file inttypes.h.

**#define SCNo64**

Definition at line 234 of file inttypes.h.

**#define SCNo8**

Definition at line 195 of file inttypes.h.

**#define SCNoFAST16**

Definition at line 216 of file inttypes.h.

**#define SCNoFAST32**

Definition at line 229 of file inttypes.h.

**#define SCNoFAST64**

Definition at line 242 of file inttypes.h.

**#define SCNoFAST8**

Definition at line 203 of file inttypes.h.

**#define SCNoLEAST16**

Definition at line 212 of file inttypes.h.

**#define SCNoLEAST32**

Definition at line 225 of file inttypes.h.

**#define SCNoLEAST64**

Definition at line 238 of file inttypes.h.

**#define SCNoLEAST8**

Definition at line 199 of file inttypes.h.

**#define SCNoMAX**

Definition at line 247 of file inttypes.h.

**#define SCNoPTR**

Definition at line 258 of file inttypes.h.

**#define SCNu16**

Definition at line 209 of file inttypes.h.

**#define SCNu32**

Definition at line 222 of file inttypes.h.

**#define SCNu64**

Definition at line 235 of file inttypes.h.

**#define SCNu8**

Definition at line 196 of file inttypes.h.

**#define SCNuFAST16**

Definition at line 217 of file inttypes.h.

**#define SCNuFAST32**

Definition at line 230 of file inttypes.h.

**#define SCNuFAST64**

Definition at line 243 of file inttypes.h.

**#define SCNuFAST8**

Definition at line 204 of file inttypes.h.

**#define SCNuLEAST16**

Definition at line 213 of file inttypes.h.

**#define SCNuLEAST32**

Definition at line 226 of file inttypes.h.

**#define SCNuLEAST64**

Definition at line 239 of file inttypes.h.

**#define SCNuLEAST8**

Definition at line 200 of file inttypes.h.

**#define SCNuMAX**

Definition at line 248 of file inttypes.h.

**#define SCNuPTR**

Definition at line 259 of file inttypes.h.

**#define SCNx16**

Definition at line 210 of file inttypes.h.

**#define SCNX16**

Definition at line 211 of file inttypes.h.

**#define SCNx32**

Definition at line 223 of file inttypes.h.

**#define SCNX32**

Definition at line 224 of file inttypes.h.

**#define SCNx64**

Definition at line 236 of file inttypes.h.

**#define SCNX64**

Definition at line 237 of file inttypes.h.

**#define SCNx8**

Definition at line 197 of file inttypes.h.

**#define SCNX8**

Definition at line 198 of file inttypes.h.

**#define SCNxFAST16**

Definition at line 218 of file inttypes.h.

**#define SCNXFAST16**

Definition at line 219 of file inttypes.h.

**#define SCNxFAST32**

Definition at line 231 of file inttypes.h.

**#define SCNXFAST32**

Definition at line 232 of file inttypes.h.

**#define SCNxFAST64**

Definition at line 244 of file inttypes.h.

**#define SCNXFAST64**

Definition at line 245 of file inttypes.h.

**#define SCNxFAST8**

Definition at line 205 of file inttypes.h.

**#define SCNXFAST8**

Definition at line 206 of file inttypes.h.

**#define SCNxLEAST16**

Definition at line 214 of file inttypes.h.



**#define SCNXLEAST16**

Definition at line 215 of file inttypes.h.

**#define SCNxLEAST32**

Definition at line 227 of file inttypes.h.

**#define SCNXLEAST32**

Definition at line 228 of file inttypes.h.

**#define SCNxLEAST64**

Definition at line 240 of file inttypes.h.

**#define SCNXLEAST64**

Definition at line 241 of file inttypes.h.

**#define SCNxLEAST8**

Definition at line 201 of file inttypes.h.

**#define SCNXLEAST8**

Definition at line 202 of file inttypes.h.

**#define SCNxMAX**

Definition at line 249 of file inttypes.h.

**#define SCNXMAX**

Definition at line 250 of file inttypes.h.

**#define SCNxPTR**

Definition at line 260 of file inttypes.h.

**#define SCNXPTR**

Definition at line 261 of file inttypes.h.

### **#define strtouimax**

Definition at line 297 of file inttypes.h.

### **#define strtoumax**

Definition at line 298 of file inttypes.h.

### **#define wcstoimax**

Definition at line 301 of file inttypes.h.

### **#define wcstoumax**

Definition at line 302 of file inttypes.h.

---

## **Function Documentation**

**imaxdiv\_t \_\_cdecl imaxdiv (intmax\_t *numer*, intmax\_t *denom*)[inline]**

Definition at line 280 of file inttypes.h.

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/log\_event.h**

### **File Reference**

```
#include "log_event.h"  
#include "event_dispatcher.h"  
#include <string>
```

### **Classes**

- class **sosicon::LogEvent**
- *Log event.* class **sosicon::LogEventDispatcher**

### **Namespaces**

- **sosicon**  
*Application root.*

**/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/logger.cpp**

**File Reference**

```
#include "logger.h"  
#include "sosi/sosi_charset_singleton.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/logger.h File Reference**

```
#include "utils.h"
#include "log_event.h"
#include "event_dispatcher.h"
#include <iostream>
#include <algorithm>
#include <sstream>
#include <string>
```

### **Classes**

- class **sosicon::Logger**

### ***SOSI logger.* Namespaces**

- **sosicon**

### ***Application root.* Functions**

- **Logger & sosicon::flush** (Logger &l)

### **Variables**

- **Logger sosicon::logstream**

## /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/main.cpp File Reference

```
#include "main.h"
```

### Functions

- `int main (int argc, char *argv[])`
- 

### Function Documentation

`int main (int argc, char * argv[])`

Definition at line 20 of file main.cpp.

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/main.h File Reference**

```
#include <exception>
#include <ios>
#include <iostream>
#include <locale>
#include "command_line.h"
#include "factory.h"
#include "logger.h"
#include "interface/i_converter.h"
```

### **Namespaces**

- **sosicon**

*Application root.*

**/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/parser.cpp**

## **File Reference**

```
#include "parser.h"
```



## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/parser.h File Reference**

```
#include <iostream>
#include <algorithm>
#include <string>
#include <sstream>
#include <vector>
#include <map>
#include "utils.h"
#include "command_line.h"
#include "sosi/sosi_element.h"
#include "sosi/sosi_charset_singleton.h"
#include "interface/i_sosi_element.h"
```

### **Classes**

- class **sosicon::Parser**

### ***SOSI file parser.* Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/parser\_ragel. cpp File Reference**

```
#include "parser.h"
```

### **Namespaces**

- **sosicon**

*Application root.*

**/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/parser.rl**  
**File Reference**

**/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/sosi\_no  
rth\_east.rl File Reference**

**/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/rage/sosi\_north\_east\_height.rl File Reference**

**/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/rage/sosi\_origo\_ne.rl File Reference**

**/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/ragel/sosi\_ref  
.rl File Reference**

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile.cpp File Reference**

```
#include "shapefile.h"
```



## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile.h File Reference**

```
#include <algorithm>
#include <ctime>
#include <string>
#include <vector>
#include <iostream>
#include "shapefile_types.h"
#include "../logger.h"
#include "../byte_order.h"
#include "../utils.h"
#include "../coordinate_collection.h"
#include "../sosi/sosi_types.h"
#include "../sosi/sosi_element.h"
#include "../sosi/sosi_element_search.h"
#include "../interface/i_shapefile.h"
#include "../interface/i_coordinate.h"
```

### **Classes**

- class **sosicon::shape::Shapefile**

### **Shapefile *implementation*. Namespaces**

- **sosicon**
- *Application root.* **sosicon::shape**

### **ESRI Shape. Functions**

- ShapeType **sosicon::shape::getShapeEquivalent** (sosi::ElementType sosiType)  
*Resolve geometry type.*

## /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/shape/shapefile\_types.h File Reference

```
#include <stdint.h>
#include <map>
#include <algorithm>
#include <limits>
```

### Classes

- union **sosicon::shape::Int8Field**
- *8 bit integer / byte field* union **sosicon::shape::Int16Field**
- *16 bit integer / byte field* union **sosicon::shape::Int32Field**
- *32 bit integer / byte field* union **sosicon::shape::Int32TField**
- *32 bit integer / byte / geom::ShapeType field* union **sosicon::shape::DoubleField**
- *32 bit double / byte field* struct **sosicon::shape::ShxIndex**

### Namespaces

- **sosicon**
- *Application root.* **sosicon::shape**

### ESRI Shape. Typedefs

- typedef std::map< std::string, std::string > **sosicon::shape::DbfRecord**
- typedef std::vector< DbfRecord > **sosicon::shape::DbfRecordSet**
- typedef std::map< std::string, int > **sosicon::shape::DbfFieldLengths**
- typedef std::vector< ShxIndex > **sosicon::shape::ShxOffsets**

### Enumerations

- enum **sosicon::shape::ShapeType** { **sosicon::shape::shape\_type\_none**, **sosicon::shape::shape\_type\_nullShape**, **sosicon::shape::shape\_type\_point**, **sosicon::shape::shape\_type\_polyLine**, **sosicon::shape::shape\_type\_polygon**, **sosicon::shape::shape\_type\_multipoint**, **sosicon::shape::shape\_type\_pointZ**, **sosicon::shape::shape\_type\_polyLineZ**, **sosicon::shape::shape\_type\_polygonZ**, **sosicon::shape::shape\_type\_multipointZ**, **sosicon::shape::shape\_type\_pointM**, **sosicon::shape::shape\_type\_polyLineM**, **sosicon::shape::shape\_type\_polygonM**, **sosicon::shape::shape\_type\_multiPointM**, **sosicon::shape::shape\_type\_multiPatch** } *Geometry types.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_charset\_singleton.cpp File Reference**

```
#include "sosi_charset_singleton.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_charset\_singleton.h File Reference**

```
#include "../interface/i_sosi_element.h"
#include "../interface/i_sosi_head_member.h"
#include "sosi_types.h"
#include <iostream>
#include <string>
#include <sstream>
#include <vector>
```

### **Classes**

- class **sosicon::sosi::SosiCharsetSingleton**

### ***SOSI Character set.* Namespaces**

- **sosicon**
- *Application root.* **sosicon::sosi**
- *SOSI.* **sosicon::sosi::chartables**

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_element.cpp File Reference**

```
#include "sosi_element.h"
```

## /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_element.h File Reference

```
#include <vector>
#include <string>
#include "../logger.h"
#include "sosi_element_search.h"
#include "sosi_translation_table.h"
#include "sosi_charset_singleton.h"
#include "sosi_types.h"
#include "../interface/i_sosi_element.h"
```

### Classes

- class **sosicon::sosi::SosiElement**

### *Basic SOSI element.* Namespaces

- **sosicon**
- *Application root.* **sosicon::sosi**

### *SOSI.* Functions

- CoordSys **sosicon::sosi::sysCodeToCoordSys** (int sysCode)  
*Convert SOSI SYSKODE value to coordinate system data.*
- ElementType **sosicon::sosi::sosiNameToType** (std::string sosiElementName)  
*Convert SOSI element names to ElementType enum value.*
- ObjType **sosicon::sosi::sosiObjNameToType** (std::string sosiObjTypeName)  
*Convert SOSI objtype names to ObjType enum value.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_element\_search.cpp File Reference**

```
#include "sosi_element_search.h"  
#include "../interface/i_sosi_element.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_element\_search.h File Reference**

```
#include <map>
#include <vector>
#include "sosi_types.h"
```

### **Classes**

- class **sosicon::sosi::SosiElementSearch**

### **Namespaces**

- **sosicon**
- *Application root.* **sosicon::sosi**

### **SOSI. Typedefs**

- typedef std::map< std::string, ISosiElement \* > **sosicon::sosi::SosiElementMap**  
*Element index type.*
- typedef std::vector< ISosiElement \* > **sosicon::sosi::SosiChildrenList**
- typedef SosiChildrenList::iterator **sosicon::sosi::SosiChildrenIterator**



## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_junction\_point.h File Reference**

```
#include "../interface/i_sosi_element.h"
#include "sosi_types.h"
#include <iostream>
#include <string>
#include <vector>
```

### **Classes**

- class **sosicon::sosi::SosiJunctionPoint**

### ***SOSI Junction point.* Namespaces**

- **sosicon**
- *Application root.* **sosicon::sosi**  
*SOSI.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_north\_east.cpp File Reference**

```
#include "sosi_north_east.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_north\_east.h File Reference**

```
#include "../logger.h"
#include "../interface/i_sosi_element.h"
#include "../interface/i_coordinate.h"
#include "../common_types.h"
#include "../coordinate.h"
#include "sosi_types.h"
#include "sosi_origo_ne.h"
#include "sosi_unit.h"
#include <algorithm>
#include <limits>
#include <string>
#include <sstream>
#include <vector>
```

### **Classes**

- class **sosicon::sosi::SosiNorthEast**

### ***SOSI North-east element.* Namespaces**

- **sosicon**
- *Application root.* **sosicon::sosi**

### ***SOSI.* Typedefs**

- typedef std::vector< SosiNorthEast \* > **sosicon::sosi::NorthEastList**  
*List of SosiNorthEast elements.*

### **Functions**

- void **sosicon::sosi::deleteNorthEast** (NorthEastList &lst)  
*Deletes **SosiNorthEast** elements of NorthEastList.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_origo\_ne.cpp File Reference**

```
#include "sosi_origo_ne.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_origo\_ne.h File Reference**

```
#include "../interface/i_sosi_element.h"
#include "../interface/i_sosi_head_member.h"
#include "sosi_types.h"
#include <iostream>
#include <sstream>
#include <string>
```

### **Classes**

- class **sosicon::sosi::SosiOrigoNE**

### ***SOSI Junction point.* Namespaces**

- **sosicon**
- *Application root.* **sosicon::sosi**  
*SOSI.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_ref\_ list.cpp File Reference**

```
#include "sosi_ref_list.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_ref\_list.h File Reference**

```
#include "../interface/i_sosi_element.h"
#include "sosi_types.h"
#include <iostream>
#include <string>
#include <vector>
```

### **Classes**

- class **sosicon::sosi::SosiRefList**

### ***SOSI REF list*. Namespaces**

- **sosicon**
- *Application root.* **sosicon::sosi**  
*SOSI.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_translation\_table.cpp File Reference**

```
#include "sosi_translation_table.h"
```



## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_translation\_table.h File Reference**

```
#include <map>
#include <vector>
#include "sosi_types.h"
#include "sosi_charset_singleton.h"
```

### **Classes**

- class **sosicon::sosi::SosiTranslationTable**

### **Namespaces**

- **sosicon**
- *Application root.* **sosicon::sosi**  
*SOSI.*

## /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_types.h File Reference

```
#include "../interface/i_coordinate.h"
#include <string>
#include <vector>
#include <map>
```

### Classes

- struct `sosicon::sosi::ReferenceData`
- *SOSI reference number.* class `sosicon::sosi::CoordSys`

### SOSI coordinate system. Namespaces

- `sosicon`
- *Application root.* `sosicon::sosi`

### SOSI. Typedefs

- typedef `std::vector< ReferenceData * >` `sosicon::sosi::GeometryRef`  
*List of SOSI references.*
- typedef `std::vector< GeometryRef * >` `sosicon::sosi::GeometryCollection`  
*Collection of SOSI reference lists.*

### Enumerations

- enum `sosicon::sosi::ElementType` { `sosicon::sosi::sosi_element_unknown`, `sosicon::sosi::sosi_element_address_identifier`, `sosicon::sosi::sosi_element_airport_roads`, `sosicon::sosi::sosi_element_airport_type`, `sosicon::sosi::sosi_element_area`, `sosicon::sosi::sosi_element_charset`, `sosicon::sosi::sosi_element_coordsys`, `sosicon::sosi::sosi_element_curve`, `sosicon::sosi::sosi_element_data_collection_date`, `sosicon::sosi::sosi_element_eof`, `sosicon::sosi::sosi_element_head`, `sosicon::sosi::sosi_element_height`, `sosicon::sosi::sosi_element_iata_code`, `sosicon::sosi::sosi_element_icao_code`, `sosicon::sosi::sosi_element_kp`, `sosicon::sosi::sosi_element_level`, `sosicon::sosi::sosi_element_max_ne`, `sosicon::sosi::sosi_element_min_ne`, `sosicon::sosi::sosi_element_municipality`, `sosicon::sosi::sosi_element_name`, `sosicon::sosi::sosi_element_ne`, `sosicon::sosi::sosi_element_neh`, `sosicon::sosi::sosi_element_objtype`, `sosicon::sosi::sosi_element_origo_ne`, `sosicon::sosi::sosi_element_owner`, `sosicon::sosi::sosi_element_point`, `sosicon::sosi::sosi_element_quality`, `sosicon::sosi::sosi_element_ref`, `sosicon::sosi::sosi_element_surface`, `sosicon::sosi::sosi_element_text`, `sosicon::sosi::sosi_element_traffic_type`, `sosicon::sosi::sosi_element_transpar`, `sosicon::sosi::sosi_element_unit`, `sosicon::sosi::sosi_element_updatedate`, `sosicon::sosi::sosi_element_water_width`, `sosicon::sosi::sosi_element_vendor`, `sosicon::sosi::sosi_element_version` } *List of SOSI element types.*
- enum `sosicon::sosi::ObjType` { `sosicon::sosi::sosi_objtype_unknown`, `sosicon::sosi::sosi_objtype_airport`, `sosicon::sosi::sosi_objtype_airport_type`, `sosicon::sosi::sosi_objtype_baseline`, `sosicon::sosi::sosi_objtype_carriageway`, `sosicon::sosi::sosi_objtype_cadastral_address`, `sosicon::sosi::sosi_objtype_coastline`, `sosicon::sosi::sosi_objtype_constituency_boundary`, `sosicon::sosi::sosi_objtype_county_boundary`, `sosicon::sosi::sosi_objtype_data_delineation`, `sosicon::sosi::sosi_objtype_edge_view`, `sosicon::sosi::sosi_objtype_fictitious_dividing_line`, `sosicon::sosi::sosi_objtype_forest`, `sosicon::sosi::sosi_objtype_developed_area`, `sosicon::sosi::sosi_objtype_golf_course`, `sosicon::sosi::sosi_objtype_industrial_area`, `sosicon::sosi::sosi_objtype_lake`, `sosicon::sosi::sosi_objtype_lane`, `sosicon::sosi::sosi_objtype_lake_edge`, `sosicon::sosi::sosi_objtype_lake_river_barrier`, `sosicon::sosi::sosi_objtype_land_use_boundary`, `sosicon::sosi::sosi_objtype_level_crossing`, `sosicon::sosi::sosi_objtype_municipal_divide`,

- sosicon::sosi::sosi\_objtype\_municipality, sosicon::sosi::sosi\_objtype\_municipality\_boundary,  
 sosicon::sosi::sosi\_objtype\_marsh, sosicon::sosi::sosi\_objtype\_national\_border,  
 sosicon::sosi::sosi\_objtype\_pedestrian\_bicycle\_road\_centre\_line,  
 sosicon::sosi::sosi\_objtype\_sea\_river\_delineation, sosicon::sosi::sosi\_objtype\_snow\_field,  
 sosicon::sosi::sosi\_objtype\_open\_land, sosicon::sosi::sosi\_objtype\_river\_brook,  
 sosicon::sosi::sosi\_objtype\_river\_brook\_edge, sosicon::sosi::sosi\_objtype\_road\_block,  
 sosicon::sosi::sosi\_objtype\_road\_centre\_line, sosicon::sosi::sosi\_objtype\_road\_under\_railway,  
 sosicon::sosi::sosi\_objtype\_sea\_surface, sosicon::sosi::sosi\_objtype\_sidewalk,  
 sosicon::sosi::sosi\_objtype\_spelling, sosicon::sosi::sosi\_objtype\_stone\_quarry,  
 sosicon::sosi::sosi\_objtype\_street\_address, sosicon::sosi::sosi\_objtype\_territorial\_boundary,  
 sosicon::sosi::sosi\_objtype\_turn\_connecting\_segment }*List of SOSI OBJTYPEs.*
- enum sosicon::sosi::Charset { sosicon::sosi::sosi\_charset\_undetermined, sosicon::sosi::sosi\_charset\_ansi,  
 sosicon::sosi::sosi\_charset\_decn7, sosicon::sosi::sosi\_charset\_dosn8,  
 sosicon::sosi::sosi\_charset\_iso8859\_1, sosicon::sosi::sosi\_charset\_iso8859\_10,  
 sosicon::sosi::sosi\_charset\_nd7, sosicon::sosi::sosi\_charset\_utf8 }*SOSI character encodings.*
  - enum sosicon::sosi::JunctionPoint { sosicon::sosi::sosi\_junction\_node,  
 sosicon::sosi::sosi\_junction\_connection, sosicon::sosi::sosi\_junction\_open\_end }*Default SOSI junction point  
 layer types.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_unit .cpp File Reference**

```
#include "sosi_unit.h"
```

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi/sosi\_unit .h File Reference**

```
#include "../interface/i_sosi_element.h"
#include "../interface/i_sosi_head_member.h"
#include "sosi_types.h"
#include <iostream>
#include <string>
#include <sstream>
#include <vector>
```

### **Classes**

- class **sosicon::sosi::SosiUnit**

### **SOSI Unit. Namespaces**

- **sosicon**
- *Application root.* **sosicon::sosi**  
*SOSI.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi\_north\_east\_height\_ragel.cpp File Reference**

```
#include "sosi/sosi_north_east.h"
```

### **Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi\_north\_east\_ragel.cpp File Reference**

```
#include "sosi/sosi_north_east.h"
```

### **Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi\_origo\_ne\_ragel.cpp File Reference**

```
#include "sosi/sosi_origo_ne.h"
```

### **Namespaces**

- **sosicon**

*Application root.*



## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/sosi\_ref\_range l.cpp File Reference**

```
#include "sosi/sosi_ref_list.h"
```

### **Namespaces**

- **sosicon**

*Application root.*

## **/Volumes/Media/Dropbox/projects/gitsource/sosicon/src/utils.cpp File Reference**

```
#include "utils.h"
```

## /Volumes/Media/Dropbox/projects/gitsource/sosicon/src/utils.h File Reference

```
#include "memory.h"
#include "common_types.h"
#include <sys/stat.h>
#include <locale>
#include <iomanip>
#include <iostream>
#include <sstream>
#include <string>
#include <vector>
#include <algorithm>
#include <ctype.h>
```

### Namespaces

- **sosicon**
- *Application root.* **sosicon::utils**

### String manipulation routines. Functions

- `std::string sosicon::utils::className2FileName (const std::string &className)`  
*Converts Class name to file name string.*
- `std::vector< std::string > sosicon::utils::explode (char delimiter, std::string str)`  
*Split a string by a character.*
- `bool sosicon::utils::fileExists (const std::string &name)`  
*Test if file exists.*
- `bool sosicon::utils::isNumeric (const std::string &str)`  
*Test if a string represents a numeric value.*
- `std::string sosicon::utils::nonExistingFilename (std::string defaultName)`  
*Asserts output file name to be non-existing.*
- `std::string sosicon::utils::normalizeAppClassName (const std::string &className)`  
*Asserts correct name of application classes.*
- `std::string sosicon::utils::purgeCrLf (std::string str)`  
*Remove carriage returns and line feeds.*
- `std::string sosicon::utils::repeat (const std::string &seq, unsigned int count)`  
*Repeat string N times.*
- `std::string sosicon::utils::replaceAll (const std::string &from, const std::string &to, const std::string &subject)`  
*Replace all occurrences of one string with another.*
- `std::string sosicon::utils::sqlNormalize (const std::string &str)`  
*Sanitizes SQL data string.*
- `std::string sosicon::utils::stripTrailingSlash (const std::string &str)`  
*Remove trailing forward- and backward slashes from path component.*
- `std::string sosicon::utils::trim (const std::string &str)`  
*Removes leading and trailing space characters.*
- `std::string sosicon::utils::trimLeft (const std::string &str)`
- `std::string sosicon::utils::trimRight (const std::string &str)`
- `std::string sosicon::utils::toFieldname (const std::string &from)`

*Substitutes Norwegian characters.*

- `std::string sosicon::utils::toLower` (`const std::string &from`)
- `std::string sosicon::utils::ucFirst` (`const std::string &str`)
- `std::string sosicon::utils::unquote` (`const std::string &str`)

*Remove quotes around string.*

- `void sosicon::utils::getPathInfo` (`std::string path`, `std::string &dir`, `std::string &tit`, `std::string &ext`)
- `std::string sosicon::utils::wktToStr` (`Wkt wktGeom`)

*Get Well Known Text from Wkt enum.*

# Index

/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/byte\_order.cpp;174  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/byte\_order.h;175  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/command\_line.cpp;176  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/command\_line.h;177  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/common\_types.h;178  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/converter\_sosi\_stat.cpp;187  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/converter\_sosi\_stat.h;188  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/converter\_sosi2psql.cpp;179  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/converter\_sosi2psql.h;180  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/converter\_sosi2shp.cpp;181  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/converter\_sosi2shp.h;182  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/converter\_sosi2tsv.cpp;183  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/converter\_sosi2tsv.h;184  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/converter\_sosi2xml.cpp;185  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/converter\_sosi2xml.h;186  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/coordinate.h;189  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/coordinate\_collection.cpp;190  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/coordinate\_collection.h;191  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/event\_dispatcher.h;192  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/factory.cpp;193  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/factory.h;194  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_binary\_streamable.h;195  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_converter.h;196  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_coordinate.h;197  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_lookup\_table.h;198  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_rectangle.h;199

/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_shape\_element.h;200  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_shape\_element\_header.h;201  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_shape\_header.h;202  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_shapefile.h;203  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_shapefile\_dbf\_part.h;204  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_shapefile\_prj\_part.h;205  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_shapefile\_shp\_part.h;206  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_shapefile\_shx\_part.h;207  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_sosi\_element.h;208  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/interface/i\_sosi\_head\_member.h;209  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/inttypes.h;210  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/log\_event.h;230  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/logger.cpp;231  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/logger.h;232  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/main.cpp;233  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/main.h;234  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/parser.cpp;235  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/parser.h;236  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/parser\_ragel.cpp;237  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/ragel/parser.rl;238  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/ragel/sosi\_north\_east.rl;239  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/ragel/sosi\_north\_east\_height.rl;240  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/ragel/sosi\_origo\_ne.rl;241  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/ragel/sosi\_ref.rl;242  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/shape/shapefile.cpp;243  
/Volumes/Media/Dropbox/projects/gitsource/sosicon/  
src/shape/shapefile.h;244

/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/shape/shapefile_types.h;245	~ConverterSosi2tsv sosicon::ConverterSosi2tsv;54
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_charset_singleton.cpp;246	~ConverterSosi2xml sosicon::ConverterSosi2xml;57
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_charset_singleton.h;247	~ConverterSosiStat sosicon::ConverterSosiStat;59
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_element.cpp;248	~Coordinate sosicon::Coordinate;62
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_element.h;249	~CoordinateCollection sosicon::CoordinateCollection;66
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_element_search.cpp;250	~IBinaryStreamable sosicon::IBinaryStreamable;80
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_element_search.h;251	~IConverter sosicon::IConverter;82
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_junction_point.h;252	~ICoordinate sosicon::ICoordinate;85
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_north_east.cpp;253	~ILookupTable sosicon::ILookupTable;87
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_north_east.h;254	~IRectangle sosicon::IRectangle;94
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_origo_ne.cpp;255	~IShapeElement sosicon::IShapeElement;97
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_origo_ne.h;256	~IShapeElementHeader sosicon::IShapeElementHeader;100
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_ref_list.cpp;257	~IShapefile sosicon::IShapefile;101
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_ref_list.h;258	~IShapeHeader sosicon::IShapeHeader;111
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_translation_table.cpp;259	~ISosiElement sosicon::ISosiElement;114
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_translation_table.h;260	~ISosiHeadMember sosicon::ISosiHeadMember;116
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_types.h;261	~Listener sosicon::EventDispatcher::Listener;118
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_unit.cpp;263	~Parser sosicon::Parser;125
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi/sosi_unit.h;264	~Shapefile sosicon::shape::Shapefile;132
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi_north_east_height_ragel.cpp;265	~SosiCharsetSingleton sosicon::sosi::SosiCharsetSingleton;145
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi_north_east_ragel.cpp;266	~SosiJunctionPoint sosicon::sosi::SosiJunctionPoint;157
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi_origo_ne_ragel.cpp;267	~SosiNorthEast sosicon::sosi::SosiNorthEast;160
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/sosi_ref_ragel.cpp;268	~SosiOrigoNE sosicon::sosi::SosiOrigoNE;165
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/utls.cpp;269	~SosiRefList sosicon::sosi::SosiRefList;168
/Volumes/Media/Dropbox/projects/gitsource/sosicon/ src/utls.h;270	~SosiUnit sosicon::sosi::SosiUnit;172
~CommandLine sosicon::CommandLine;36	addChild sosicon::ISosiElement;114 sosicon::sosi::SosiElement;149
~ConverterSosi2psql sosicon::ConverterSosi2psql;44	addEventListener sosicon::EventDispatcher;75 sosicon::Logger;122
~ConverterSosi2shp sosicon::ConverterSosi2shp;51	

adjustMasterMbr	sosicon::shape::Shapefile;132
append	sosicon::sosi::SosiNorthEast;160
b	sosicon::shape::DoubleField;73
	sosicon::shape::Int16Field;90
	sosicon::shape::Int32Field;91
	sosicon::shape::Int32TField;92
	sosicon::shape::Int8Field;93
back	sosicon::sosi::SosiNorthEast;160
big	sosicon::byteOrder;22
bottom	sosicon::IRectangle;95
build	sosicon::IShapefile;102
	sosicon::shape::Shapefile;132
buildCreateStatement	sosicon::ConverterSosi2psql;44
buildCreateStatements	sosicon::ConverterSosi2psql;44
buildDbf	sosicon::shape::Shapefile;132
buildDbfFieldDescriptor	sosicon::shape::Shapefile;133
buildDbfHeader	sosicon::shape::Shapefile;133
buildDbfRecordSection	sosicon::shape::Shapefile;133
buildInsertStatement	sosicon::ConverterSosi2psql;45
buildInsertStatements	sosicon::ConverterSosi2psql;45
buildShpElement	sosicon::shape::Shapefile;134
buildShpHeader	sosicon::shape::Shapefile;134
buildShpPoint	sosicon::shape::Shapefile;134
buildShpPolygon	sosicon::shape::Shapefile;134
buildShpPolyLine	sosicon::shape::Shapefile;135
buildShpRecCoordinate	sosicon::shape::Shapefile;135
buildShpRecCoordinates	sosicon::shape::Shapefile;135
buildShpRecHeaderCommonPart	sosicon::shape::Shapefile;136
buildShpRecHeaderExtended	sosicon::shape::Shapefile;136
buildShpRecHeaderOffsets	sosicon::shape::Shapefile;136
buildShx	sosicon::shape::Shapefile;137
byte_order.cpp	sosicon;174
Charset	SOSI Elements;13
children	sosicon::ISosiElement;114
	sosicon::sosi::SosiElement;150
className2FileName	sosicon::utils;31
cleanup	sosicon::ConverterSosi2psql;45; 46
CommandLine	sosicon::CommandLine;36
complete	sosicon::Parser;125
Converters;10	
ConverterSosi2psql	sosicon::ConverterSosi2psql;44
ConverterSosi2shp	sosicon::ConverterSosi2shp;51
ConverterSosi2tsv	sosicon::ConverterSosi2tsv;54
ConverterSosi2xml	sosicon::ConverterSosi2xml;56
ConverterSosiStat	sosicon::ConverterSosiStat;59
Coordinate	sosicon::Coordinate;62
CoordinateCollection	sosicon::CoordinateCollection;66
CoordinateList	sosicon;19
CoordSys	sosicon::sosi::CoordSys;71
d	sosicon::shape::DoubleField;73
DbfFieldLengths	sosicon::shape;25
DbfRecord	sosicon::shape;25
DbfRecordSet	sosicon::shape;25
deleteChildren	sosicon::ISosiElement;114
	sosicon::sosi::SosiElement;150
deleteNorthEast	sosicon::sosi;28
determine	sosicon::byteOrder;22
digestPendingElement	sosicon::Parser;125
discoverCoords	sosicon::CoordinateCollection;66
Dispatch	sosicon::EventDispatcher;75

- displayString
  - sosicon::sosi::CoordSys;71
- divide
  - sosicon::Coordinate;62
  - sosicon::ICoordinate;85
- doubleToLittleEndian
  - sosicon::byteOrder;23
- dump
  - sosicon::ISosiElement;114
  - sosicon::Parser;126
  - sosicon::sosi::SosiElement;150
  - sosicon::sosi::SosiNorthEast;161
- element
  - sosicon::sosi::SosiElementSearch;155
- ElementType
  - SOSI Elements;14
- endianness
  - sosicon::byteOrder;24
- Endianness
  - sosicon::byteOrder;22
- equals
  - sosicon::Coordinate;62
  - sosicon::ICoordinate;85
- expand
  - sosicon::ConverterSosi2psql::Field;78
- expandBoundingBox
  - sosicon::sosi::SosiNorthEast;161
- expandShpBuffer
  - sosicon::shape::Shapefile;137
- explode
  - sosicon::utils;31
- extractData
  - sosicon::ConverterSosi2psql;46
- extractDbfFields
  - sosicon::shape::Shapefile;137
- extractPath
  - sosicon::CoordinateCollection;66
- Field
  - sosicon::ConverterSosi2psql::Field;78
- FieldsList
  - sosicon::ConverterSosi2psql;43
- FieldsListCollection
  - sosicon::ConverterSosi2psql;43
- fileExists
  - sosicon::utils;31
- filterSosiId
  - sosicon::IShapefile;102
  - sosicon::shape::Shapefile;137
- find
  - sosicon::ISosiElement;114
  - sosicon::sosi::SosiElement;150
- flush
  - sosicon;20
- free
  - sosicon::CoordinateCollection;67
  - sosicon::sosi::SosiNorthEast;161
- front
  - sosicon::sosi::SosiNorthEast;161
- GeometryCollection
  - SOSI Elements;13
- GeometryRef
  - SOSI Elements;13
- get
  - sosicon::Factory;76
  - sosicon::ILookupTable;87
- getBoundingBox
  - sosicon::IShapeHeader;112
- getByteSize
  - sosicon::IShapeElement;98
  - sosicon::IShapeHeader;112
- getChild
  - sosicon::ISosiElement;114
  - sosicon::sosi::SosiElement;150
- getData
  - sosicon::ISosiElement;114
  - sosicon::sosi::SosiElement;150
- getDivisor
  - sosicon::sosi::SosiUnit;173
- getE
  - sosicon::Coordinate;62
  - sosicon::ICoordinate;85
  - sosicon::sosi::SosiOrigoNE;165
- getEncoding
  - sosicon::sosi::SosiCharsetSingleton;145
- getEncodingName
  - sosicon::sosi::SosiCharsetSingleton;145
- getFileLength
  - sosicon::IShapeHeader;112
- getGeom
  - sosicon::CoordinateCollection;67
- getGeomSizes
  - sosicon::CoordinateCollection;67
- getHoles
  - sosicon::CoordinateCollection;67
- getHoleSizes
  - sosicon::CoordinateCollection;67
- getInstance
  - sosicon::sosi::SosiCharsetSingleton;145
- getLevel
  - sosicon::ISosiElement;114
  - sosicon::sosi::SosiElement;150
- getMBR
  - sosicon::IShapeElement;98
- getN
  - sosicon::Coordinate;62
  - sosicon::ICoordinate;85
  - sosicon::sosi::SosiOrigoNE;165
- getName
  - sosicon::ISosiElement;114
  - sosicon::sosi::SosiElement;151
- getNext
  - sosicon;20





PRIIdLEAST8;214  
PRIIdMAX;214  
PRIIdPTR;214  
PRIi16;215  
PRIi32;215  
PRIi64;215  
PRIi8;215  
PRIiFAST16;215  
PRIiFAST32;215  
PRIiFAST64;215  
PRIiFAST8;215  
PRIiLEAST16;215  
PRIiLEAST32;215  
PRIiLEAST64;215  
PRIiLEAST8;216  
PRIiMAX;216  
PRIiPTR;216  
PRIo16;216  
PRIo32;216  
PRIo64;216  
PRIo8;216  
PRIoFAST16;216  
PRIoFAST32;216  
PRIoFAST64;216  
PRIoFAST8;216  
PRIoLEAST16;217  
PRIoLEAST32;217  
PRIoLEAST64;217  
PRIoLEAST8;217  
PRIoMAX;217  
PRIoPTR;217  
PRIu16;217  
PRIu32;217  
PRIu64;217  
PRIu8;217  
PRIuFAST16;217  
PRIuFAST32;218  
PRIuFAST64;218  
PRIuFAST8;218  
PRIuLEAST16;218  
PRIuLEAST32;218  
PRIuLEAST64;218  
PRIuLEAST8;218  
PRIuMAX;218  
PRIuPTR;218  
PRIx16;218  
PRIX16;218  
PRIx32;219  
PRIX32;219  
PRIx64;219  
PRIX64;219  
PRIx8;219  
PRIX8;219  
PRIxFAST16;219  
PRIXFAST16;219  
PRIxFAST32;219

PRIXFAST32;219  
PRIxFAST64;219  
PRIXFAST64;220  
PRIxFAST8;220  
PRIXFAST8;220  
PRIxLEAST16;220  
PRIXLEAST16;220  
PRIxLEAST32;220  
PRIXLEAST32;220  
PRIxLEAST64;220  
PRIXLEAST64;220  
PRIxLEAST8;220  
PRIXLEAST8;220  
PRIxMAX;221  
PRIXMAX;221  
PRIxPTR;221  
PRIXPTR;221  
SCNd16;221  
SCNd32;221  
SCNd64;221  
SCNd8;221  
SCNdFAST16;221  
SCNdFAST32;221  
SCNdFAST64;221  
SCNdFAST8;222  
SCNdLEAST16;222  
SCNdLEAST32;222  
SCNdLEAST64;222  
SCNdLEAST8;222  
SCNdMAX;222  
SCNdPTR;222  
SCNi16;222  
SCNi32;222  
SCNi64;222  
SCNi8;222  
SCNiFAST16;223  
SCNiFAST32;223  
SCNiFAST64;223  
SCNiFAST8;223  
SCNiLEAST16;223  
SCNiLEAST32;223  
SCNiLEAST64;223  
SCNiLEAST8;223  
SCNiMAX;223  
SCNiPTR;223  
SCNo16;223  
SCNo32;224  
SCNo64;224  
SCNo8;224  
SCNoFAST16;224  
SCNoFAST32;224  
SCNoFAST64;224  
SCNoFAST8;224  
SCNoLEAST16;224  
SCNoLEAST32;224  
SCNoLEAST64;224

SCNoLEAST8;224	JunctionPoint
SCNoMAX;225	SOSI Elements;15
SCNoPTR;225	left
SCNu16;225	sosicon::IRectangle;95
SCNu32;225	leftOf
SCNu64;225	sosicon::Coordinate;63
SCNu8;225	sosicon::ICoordinate;85
SCNuFAST16;225	length
SCNuFAST32;225	sosicon::ConverterSosi2psql::Field;78
SCNuFAST64;225	sosicon::shape::ShxIndex;143
SCNuFAST8;225	ListenerLst
SCNuLEAST16;225	sosicon::EventDispatcher;74
SCNuLEAST32;226	little
SCNuLEAST64;226	sosicon::byteOrder;22
SCNuLEAST8;226	LogEvent
SCNuMAX;226	sosicon::LogEvent;119
SCNuPTR;226	logstream
SCNx16;226	sosicon;21
SCNX16;226	main
SCNx32;226	main.cpp;233
SCNX32;226	main.cpp
SCNx64;226	main;233
SCNX64;226	makeBasePath
SCNx8;227	sosicon::ConverterSosi2shp;51
SCNX8;227	makePsql
SCNxFAST16;227	sosicon::ConverterSosi2psql;48
SCNXFAST16;227	makeShp
SCNxFAST32;227	sosicon::ConverterSosi2shp;51
SCNXFAST32;227	makeStat
SCNxFAST64;227	sosicon::ConverterSosiStat;59
SCNXFAST64;227	makeXML
SCNxFAST8;227	sosicon::ConverterSosi2xml;57
SCNXFAST8;227	mAltitude
SCNxLEAST16;227	sosicon::Coordinate;64
SCNXLEAST16;228	mAppend
SCNxLEAST32;228	sosicon::CommandLine;38
SCNXLEAST32;228	matchTypes
SCNxLEAST64;228	sosicon::sosi::SosiElementSearch;155
SCNXLEAST64;228	MAX_BUFFER_CHUNK_SIZE
SCNxLEAST8;228	sosicon::shape::Shapefile;139
SCNXLEAST8;228	MAX_COORDSYS_TABLE
SCNxMAX;228	sosicon::sosi::SosiTranslationTable;170
SCNXMAX;228	mCharset
SCNxPTR;228	sosicon::sosi::SosiCharsetSingleton;146
SCNXPTR;228	mCharsetName
strtoimax;229	sosicon::sosi::SosiCharsetSingleton;146
strtoumax;229	mChildren
wcstoimax;229	sosicon::sosi::SosiElement;152
wcstoumax;229	mCmd
isClockwise	sosicon::ConverterSosi2psql;49
sosicon;20	sosicon::ConverterSosi2shp;52
isCounterClockwise	sosicon::ConverterSosi2tsv;55
sosicon;20	sosicon::ConverterSosi2xml;57
isNumeric	sosicon::ConverterSosiStat;60
sosicon::ConverterSosi2psql::Field;78	mCommand
sosicon::utils;31	sosicon::CommandLine;38

- mCoordinates
  - sosicon::sosi::SosiNorthEast;162
- mCoordinatesIterator
  - sosicon::sosi::SosiNorthEast;162
- mCoordSysTable
  - sosicon::sosi::SosiTranslationTable;170
- mCreateStatements
  - sosicon::CommandLine;38
- mCurrentCharset
  - sosicon::Parser;126
- mCurrentSourcefile
  - sosicon::ConverterSosi2psql;49
  - sosicon::ConverterSosi2shp;52
- mData
  - sosicon::sosi::SosiElement;152
- mDbfBuffer
  - sosicon::shape::Shapefile;139
- mDbfBufferSize
  - sosicon::shape::Shapefile;139
- mDbfFieldLengths
  - sosicon::shape::Shapefile;140
- mDbfHeader
  - sosicon::shape::Shapefile;140
- mDbfRecordSet
  - sosicon::shape::Shapefile;140
- mDbSchema
  - sosicon::CommandLine;38
- mDbTable
  - sosicon::CommandLine;38
- mDestinationDirectory
  - sosicon::CommandLine;38
- mDisplayString
  - sosicon::sosi::CoordSys;72
- mDivisor
  - sosicon::sosi::SosiUnit;173
- mEast
  - sosicon::Coordinate;64
- mElementIndex
  - sosicon::Parser;126
- mElementStack
  - sosicon::Parser;126
- mElementTypes
  - sosicon::sosi::SosiElementSearch;156
- mFieldSelection
  - sosicon::CommandLine;39
- mFieldsListCollection
  - sosicon::ConverterSosi2psql;49
- mFilterSosiId
  - sosicon::CommandLine;39
  - sosicon::shape::Shapefile;140
- mFilterSosiObjTypes
  - sosicon::shape::Shapefile;140
- mGeom
  - sosicon::CoordinateCollection;68
- mGeomIndex
  - sosicon::CoordinateCollection;68
- mGeomNormalized
  - sosicon::CoordinateCollection;68
- mGeomSizes
  - sosicon::CoordinateCollection;68
- mGeomTypes
  - sosicon::CommandLine;39
- mGeoTypes
  - sosicon::ConverterSosiStat;60
- mHoles
  - sosicon::CoordinateCollection;69
- mHoleSizes
  - sosicon::CoordinateCollection;69
- mHolesNormalized
  - sosicon::CoordinateCollection;69
- mIncludeHeader
  - sosicon::CommandLine;39
- mIndex
  - sosicon::sosi::SosiElement;152
  - sosicon::sosi::SosiElementSearch;156
- mInitialized
  - sosicon::sosi::SosiCharsetSingleton;146
  - sosicon::sosi::SosiOrigoNE;165
  - sosicon::sosi::SosiUnit;173
- mInsertStatements
  - sosicon::CommandLine;39
- mInstance
  - sosicon::sosi::SosiCharsetSingleton;146
- mIsNumeric
  - sosicon::ConverterSosi2psql::Field;79
- mIsTtyIn
  - sosicon::CommandLine;39
- mIsTtyOut
  - sosicon::CommandLine;40
- mLevel
  - sosicon::sosi::SosiElement;152
- mListeners
  - sosicon::EventDispatcher;75
- mLogEventDispatcher
  - sosicon::Logger;123
- mMakeSubDir
  - sosicon::CommandLine;40
- mMaxLength
  - sosicon::ConverterSosi2psql::Field;79
- mMaxX
  - sosicon::sosi::SosiNorthEast;162
- mMaxY
  - sosicon::sosi::SosiNorthEast;162
- mMessage
  - sosicon::LogEvent;119
- mMinLength
  - sosicon::ConverterSosi2psql::Field;79
- mMinX
  - sosicon::sosi::SosiNorthEast;162
- mMinY
  - sosicon::sosi::SosiNorthEast;162
- mMsgStream



sosicon::shape::Shapefile;142	PRIdFAST8
neListToCoordList	inttypes.h;214
sosicon;21	PRIdLEAST16
next	inttypes.h;214
sosicon::sosi::SosiElementSearch;155	PRIdLEAST32
nextChild	inttypes.h;214
sosicon::sosi::SosiElement;151	PRIdLEAST64
nonExistingFilename	inttypes.h;214
sosicon::utils;32	PRIdLEAST8
normalizeAppClassName	inttypes.h;214
sosicon::utils;32	PRIdMAX
NorthEastList	inttypes.h;214
sosicon::sosi;28	PRIdPTR
not_set	inttypes.h;214
sosicon::byteOrder;22	PRi16
ObjType	inttypes.h;215
SOSI Elements;15	PRi32
objTypeExcluded	inttypes.h;215
sosicon::ConverterSosi2psql;48	PRi64
offset	inttypes.h;215
sosicon::shape::ShxIndex;143	PRi8
onEvent	inttypes.h;215
sosicon::EventDispatcher::Listener;118	PRiFAST16
operator/=	inttypes.h;215
sosicon::sosi::SosiNorthEast;161	PRiFAST32
operator+=	inttypes.h;215
sosicon::sosi::SosiNorthEast;161	PRiFAST64
operator<<	inttypes.h;215
Interfaces;11	PRiFAST8
sosicon::Logger;122; 123	inttypes.h;215
outputDisclaimer	PRiLEAST16
sosicon::CommandLine;37	inttypes.h;215
outputHelpText	PRiLEAST32
sosicon::CommandLine;37	inttypes.h;215
outputLicense	PRiLEAST64
sosicon::CommandLine;37	inttypes.h;215
parse	PRiLEAST8
sosicon::CommandLine;37	inttypes.h;216
Parser	PRiMAX
sosicon::Parser;125	inttypes.h;216
populate	PRiPTR
sosicon::IShapeElement;98	inttypes.h;216
PRId16	printElementData
inttypes.h;213	sosicon::ConverterSosiStat;59
PRId32	printListContent
inttypes.h;213	sosicon::ConverterSosiStat;59
PRId64	printTableHeader
inttypes.h;213	sosicon::ConverterSosiStat;60
PRId8	PRIo16
inttypes.h;214	inttypes.h;216
PRIdFAST16	PRIo32
inttypes.h;214	inttypes.h;216
PRIdFAST32	PRIo64
inttypes.h;214	inttypes.h;216
PRIdFAST64	PRIo8
inttypes.h;214	inttypes.h;216

PRIoFAST16	
inttypes.h;216	
PRIoFAST32	
inttypes.h;216	
PRIoFAST64	
inttypes.h;216	
PRIoFAST8	
inttypes.h;216	
PRIoLEAST16	
inttypes.h;217	
PRIoLEAST32	
inttypes.h;217	
PRIoLEAST64	
inttypes.h;217	
PRIoLEAST8	
inttypes.h;217	
PRIoMAX	
inttypes.h;217	
PRIoPTR	
inttypes.h;217	
PRIo16	
inttypes.h;217	
PRIo32	
inttypes.h;217	
PRIo64	
inttypes.h;217	
PRIo8	
inttypes.h;217	
PRIoFAST16	
inttypes.h;217	
PRIoFAST32	
inttypes.h;218	
PRIoFAST64	
inttypes.h;218	
PRIoFAST8	
inttypes.h;218	
PRIoLEAST16	
inttypes.h;218	
PRIoLEAST32	
inttypes.h;218	
PRIoLEAST64	
inttypes.h;218	
PRIoLEAST8	
inttypes.h;218	
PRIoMAX	
inttypes.h;218	
PRIoPTR	
inttypes.h;218	
PRIx16	
inttypes.h;218	
PRIX16	
inttypes.h;218	
PRIx32	
inttypes.h;219	
PRIX32	
inttypes.h;219	
PRIx64	
inttypes.h;219	
PRIX64	
inttypes.h;219	
PRIx8	
inttypes.h;219	
PRIX8	
inttypes.h;219	
PRIxFAST16	
inttypes.h;219	
PRIXFAST16	
inttypes.h;219	
PRIxFAST32	
inttypes.h;219	
PRIXFAST32	
inttypes.h;219	
PRIxFAST64	
inttypes.h;219	
PRIXFAST64	
inttypes.h;220	
PRIxFAST8	
inttypes.h;220	
PRIXFAST8	
inttypes.h;220	
PRIxLEAST16	
inttypes.h;220	
PRIXLEAST16	
inttypes.h;220	
PRIxLEAST32	
inttypes.h;220	
PRIXLEAST32	
inttypes.h;220	
PRIxLEAST64	
inttypes.h;220	
PRIXLEAST64	
inttypes.h;220	
PRIxLEAST8	
inttypes.h;220	
PRIXLEAST8	
inttypes.h;220	
PRIxMAX	
inttypes.h;221	
PRIXMAX	
inttypes.h;221	
PRIxPTR	
inttypes.h;221	
PRIXPTR	
inttypes.h;221	
prjString	
sosicon::sosi::CoordSys;72	
purgeCrLf	
sosicon::utils;32	
quot	
imaxdiv_t;89	
ragelParseCoordinatesNe	
sosicon::sosi::SosiNorthEast;161	

ragelParseCoordinatesNeh	SCNdFAST8
sosicon::sosi::SosiNorthEast;162	inttypes.h;222
ragelParseSosiLine	SCNdLEAST16
sosicon::Parser;126	inttypes.h;222
ragelParseSosiOrigoNE	SCNdLEAST32
sosicon::sosi::SosiOrigoNE;165	inttypes.h;222
ragelParseSosiRef	SCNdLEAST64
sosicon::sosi::SosiRefList;168	inttypes.h;222
release	SCNdLEAST8
sosicon::Factory;76	inttypes.h;222
rem	SCNdMAX
imaxdiv_t;89	inttypes.h;222
removeEventListener	SCNdPTR
sosicon::EventDispatcher;75	inttypes.h;222
sosicon::Logger;123	SCNi16
repeat	inttypes.h;222
sosicon::utils;32	SCNi32
replaceAll	inttypes.h;222
sosicon::utils;33	SCNi64
reverse	inttypes.h;222
sosicon::sosi::ReferenceData;128	SCNi8
sosicon::sosi::SosiNorthEast;162	inttypes.h;222
reverseLookup	SCNiFAST16
sosicon::sosi::SosiTranslationTable;170	inttypes.h;223
right	SCNiFAST32
sosicon::IRectangle;95	inttypes.h;223
rightOf	SCNiFAST64
sosicon::Coordinate;63	inttypes.h;223
sosicon::ICoordinate;85	SCNiFAST8
RowsList	inttypes.h;223
sosicon::ConverterSosi2psql;43	SCNiLEAST16
RowsListCollection	inttypes.h;223
sosicon::ConverterSosi2psql;44	SCNiLEAST32
run	inttypes.h;223
sosicon::ConverterSosi2psql;48	SCNiLEAST64
sosicon::ConverterSosi2shp;51	inttypes.h;223
sosicon::ConverterSosi2tsv;55	SCNiLEAST8
sosicon::ConverterSosi2xml;57	inttypes.h;223
sosicon::ConverterSosiStat;60	SCNiMAX
sosicon::IConverter;83	inttypes.h;223
saveToDbf	SCNiPTR
sosicon::shape::Shapefile;138	inttypes.h;223
SCNd16	SCNo16
inttypes.h;221	inttypes.h;223
SCNd32	SCNo32
inttypes.h;221	inttypes.h;224
SCNd64	SCNo64
inttypes.h;221	inttypes.h;224
SCNd8	SCNo8
inttypes.h;221	inttypes.h;224
SCNdFAST16	SCNoFAST16
inttypes.h;221	inttypes.h;224
SCNdFAST32	SCNoFAST32
inttypes.h;221	inttypes.h;224
SCNdFAST64	SCNoFAST64
inttypes.h;221	inttypes.h;224



SCNoFAST8  
     inttypes.h;224  
 SCNoLEAST16  
     inttypes.h;224  
 SCNoLEAST32  
     inttypes.h;224  
 SCNoLEAST64  
     inttypes.h;224  
 SCNoLEAST8  
     inttypes.h;224  
 SCNoMAX  
     inttypes.h;225  
 SCNoPTR  
     inttypes.h;225  
 SCNu16  
     inttypes.h;225  
 SCNu32  
     inttypes.h;225  
 SCNu64  
     inttypes.h;225  
 SCNu8  
     inttypes.h;225  
 SCNuFAST16  
     inttypes.h;225  
 SCNuFAST32  
     inttypes.h;225  
 SCNuFAST64  
     inttypes.h;225  
 SCNuFAST8  
     inttypes.h;225  
 SCNuLEAST16  
     inttypes.h;225  
 SCNuLEAST32  
     inttypes.h;226  
 SCNuLEAST64  
     inttypes.h;226  
 SCNuLEAST8  
     inttypes.h;226  
 SCNuMAX  
     inttypes.h;226  
 SCNuPTR  
     inttypes.h;226  
 SCNx16  
     inttypes.h;226  
 SCNx16  
     inttypes.h;226  
 SCNx32  
     inttypes.h;226  
 SCNx32  
     inttypes.h;226  
 SCNx64  
     inttypes.h;226  
 SCNx64  
     inttypes.h;226  
 SCNx8  
     inttypes.h;227

SCNx8  
     inttypes.h;227  
 SCNxFAST16  
     inttypes.h;227  
 SCNxFAST16  
     inttypes.h;227  
 SCNxFAST32  
     inttypes.h;227  
 SCNxFAST32  
     inttypes.h;227  
 SCNxFAST64  
     inttypes.h;227  
 SCNxFAST64  
     inttypes.h;227  
 SCNxFAST8  
     inttypes.h;227  
 SCNxFAST8  
     inttypes.h;227  
 SCNxLEAST16  
     inttypes.h;227  
 SCNxLEAST16  
     inttypes.h;228  
 SCNxLEAST32  
     inttypes.h;228  
 SCNxLEAST32  
     inttypes.h;228  
 SCNxLEAST64  
     inttypes.h;228  
 SCNxLEAST64  
     inttypes.h;228  
 SCNxLEAST8  
     inttypes.h;228  
 SCNxLEAST8  
     inttypes.h;228  
 SCNxMAX  
     inttypes.h;228  
 SCNxMAX  
     inttypes.h;228  
 SCNxPTR  
     inttypes.h;228  
 SCNxPTR  
     inttypes.h;228  
 serial  
     sosicon::sosi::ReferenceData;128  
 setBoundingBox  
     sosicon::IShapeHeader;112  
 setE  
     sosicon::Coordinate;63  
     sosicon::ICoordinate;86  
 setFileLength  
     sosicon::IShapeHeader;112  
 setH  
     sosicon::Coordinate;63  
     sosicon::ICoordinate;86  
 setN  
     sosicon::Coordinate;63

- sosicon::ICoordinate;86
- setShapeType
  - sosicon::IShapeHeader;112
- shape\_type\_multiPatch
  - sosicon::shape;26
- shape\_type\_multipoint
  - sosicon::shape;26
- shape\_type\_multiPointM
  - sosicon::shape;26
- shape\_type\_multipointZ
  - sosicon::shape;26
- shape\_type\_none
  - sosicon::shape;26
- shape\_type\_nullShape
  - sosicon::shape;26
- shape\_type\_point
  - sosicon::shape;26
- shape\_type\_pointM
  - sosicon::shape;26
- shape\_type\_pointZ
  - sosicon::shape;26
- shape\_type\_polygon
  - sosicon::shape;26
- shape\_type\_polygonM
  - sosicon::shape;26
- shape\_type\_polygonZ
  - sosicon::shape;26
- shape\_type\_polyLine
  - sosicon::shape;26
- shape\_type\_polyLineM
  - sosicon::shape;26
- shape\_type\_polyLineZ
  - sosicon::shape;26
- Shapefile
  - sosicon::shape::Shapefile;132
- ShapeType
  - sosicon::shape;26
- shift
  - sosicon::Coordinate;63
  - sosicon::ICoordinate;86
- ShxOffsets
  - sosicon::shape;26
- SOSI Elements;11
  - Charset;13
  - ElementType;14
  - GeometryCollection;13
  - GeometryRef;13
  - JunctionPoint;15
  - ObjType;15
  - sosi\_charset\_ansi;14
  - sosi\_charset\_decn7;14
  - sosi\_charset\_dosn8;14
  - sosi\_charset\_iso8859\_1;14
  - sosi\_charset\_iso8859\_10;14
  - sosi\_charset\_nd7;14
  - sosi\_charset\_undetermined;13

- sosi\_charset\_utf8;14
- sosi\_element\_address\_identifier;14
- sosi\_element\_airport\_roads;14
- sosi\_element\_airport\_type;14
- sosi\_element\_area;14
- sosi\_element\_charset;14
- sosi\_element\_coordsys;14
- sosi\_element\_curve;14
- sosi\_element\_data\_collection\_date;14
- sosi\_element\_eof;14
- sosi\_element\_head;14
- sosi\_element\_height;14
- sosi\_element\_iata\_code;14
- sosi\_element\_icao\_code;14
- sosi\_element\_kp;14
- sosi\_element\_level;14
- sosi\_element\_max\_ne;14
- sosi\_element\_min\_ne;14
- sosi\_element\_municipality;14
- sosi\_element\_name;14
- sosi\_element\_ne;14
- sosi\_element\_neh;14
- sosi\_element\_objtype;14
- sosi\_element\_origo\_ne;14
- sosi\_element\_owner;14
- sosi\_element\_point;15
- sosi\_element\_quality;15
- sosi\_element\_ref;15
- sosi\_element\_surface;15
- sosi\_element\_text;15
- sosi\_element\_traffic\_type;15
- sosi\_element\_transpar;15
- sosi\_element\_unit;15
- sosi\_element\_unknown;14
- sosi\_element\_updatedate;15
- sosi\_element\_vendor;15
- sosi\_element\_version;15
- sosi\_element\_water\_width;15
- sosi\_junction\_connection;15
- sosi\_junction\_node;15
- sosi\_junction\_open\_end;15
- sosi\_objtype\_airport;15
- sosi\_objtype\_airport\_type;15
- sosi\_objtype\_baseline;15
- sosi\_objtype\_cadastral\_address;15
- sosi\_objtype\_carriageway;15
- sosi\_objtype\_coastline;15
- sosi\_objtype\_constituency\_boundary;15
- sosi\_objtype\_county\_boundary;15
- sosi\_objtype\_data\_delineation;15
- sosi\_objtype\_developed\_area;16
- sosi\_objtype\_edge\_view;15
- sosi\_objtype\_fictitious\_dividing\_line;16
- sosi\_objtype\_forest;16
- sosi\_objtype\_golf\_course;16
- sosi\_objtype\_industrial\_area;16

sosi_objtype_lake;16	SOSI Elements;14
sosi_objtype_lake_edge;16	sosi_element_area
sosi_objtype_lake_river_barrier;16	SOSI Elements;14
sosi_objtype_land_use_boundary;16	sosi_element_charset
sosi_objtype_lane;16	SOSI Elements;14
sosi_objtype_level_crossing;16	sosi_element_coordsys
sosi_objtype_marsh;16	SOSI Elements;14
sosi_objtype_municipal_divide;16	sosi_element_curve
sosi_objtype_municipality;16	SOSI Elements;14
sosi_objtype_municipality_boundary;16	sosi_element_data_collection_date
sosi_objtype_national_border;16	SOSI Elements;14
sosi_objtype_open_land;16	sosi_element_eof
sosi_objtype_pedestrian_bicycle_road_centre_line;	SOSI Elements;14
16	sosi_element_head
sosi_objtype_river_brook;16	SOSI Elements;14
sosi_objtype_river_brook_edge;16	sosi_element_height
sosi_objtype_road_block;16	SOSI Elements;14
sosi_objtype_road_centre_line;16	sosi_element_iata_code
sosi_objtype_road_under_railway;16	SOSI Elements;14
sosi_objtype_sea_river_delineation;16	sosi_element_icao_code
sosi_objtype_sea_surface;16	SOSI Elements;14
sosi_objtype_sidewalk;16	sosi_element_kp
sosi_objtype_snow_field;16	SOSI Elements;14
sosi_objtype_spelling;16	sosi_element_level
sosi_objtype_stone_quarry;16	SOSI Elements;14
sosi_objtype_street_address;16	sosi_element_max_ne
sosi_objtype_territorial_boundary;16	SOSI Elements;14
sosi_objtype_turn_connecting_segment;16	sosi_element_min_ne
sosi_objtype_unknown;15	SOSI Elements;14
SosiChildrenIterator;13	sosi_element_municipality
SosiChildrenList;13	SOSI Elements;14
SosiElementMap;13	sosi_element_name
sosiNameToType;17	SOSI Elements;14
sosiObjNameToType;17	sosi_element_ne
sysCodeToCoordSys;17	SOSI Elements;14
sosi_charset_ansi	sosi_element_neh
SOSI Elements;14	SOSI Elements;14
sosi_charset_decn7	sosi_element_objtype
SOSI Elements;14	SOSI Elements;14
sosi_charset_dosn8	sosi_element_origo_ne
SOSI Elements;14	SOSI Elements;14
sosi_charset_iso8859_1	sosi_element_owner
SOSI Elements;14	SOSI Elements;14
sosi_charset_iso8859_10	sosi_element_point
SOSI Elements;14	SOSI Elements;15
sosi_charset_nd7	sosi_element_quality
SOSI Elements;14	SOSI Elements;15
sosi_charset_undetermined	sosi_element_ref
SOSI Elements;13	SOSI Elements;15
sosi_charset_utf8	sosi_element_surface
SOSI Elements;14	SOSI Elements;15
sosi_element_address_identifier	sosi_element_text
SOSI Elements;14	SOSI Elements;15
sosi_element_airport_roads	sosi_element_traffic_type
SOSI Elements;14	SOSI Elements;15
sosi_element_airport_type	sosi_element_transpar

SOSI Elements;15  
 sosi\_element\_unit  
 SOSI Elements;15  
 sosi\_element\_unknown  
 SOSI Elements;14  
 sosi\_element\_updatedate  
 SOSI Elements;15  
 sosi\_element\_vendor  
 SOSI Elements;15  
 sosi\_element\_version  
 SOSI Elements;15  
 sosi\_element\_water\_width  
 SOSI Elements;15  
 sosi\_junction\_connection  
 SOSI Elements;15  
 sosi\_junction\_node  
 SOSI Elements;15  
 sosi\_junction\_open\_end  
 SOSI Elements;15  
 sosi\_objtype\_airport  
 SOSI Elements;15  
 sosi\_objtype\_airport\_type  
 SOSI Elements;15  
 sosi\_objtype\_baseline  
 SOSI Elements;15  
 sosi\_objtype\_cadastral\_address  
 SOSI Elements;15  
 sosi\_objtype\_carriageway  
 SOSI Elements;15  
 sosi\_objtype\_coastline  
 SOSI Elements;15  
 sosi\_objtype\_constituency\_boundary  
 SOSI Elements;15  
 sosi\_objtype\_county\_boundary  
 SOSI Elements;15  
 sosi\_objtype\_data\_delineation  
 SOSI Elements;15  
 sosi\_objtype\_developed\_area  
 SOSI Elements;16  
 sosi\_objtype\_edge\_view  
 SOSI Elements;15  
 sosi\_objtype\_fictitious\_dividing\_line  
 SOSI Elements;16  
 sosi\_objtype\_forest  
 SOSI Elements;16  
 sosi\_objtype\_golf\_course  
 SOSI Elements;16  
 sosi\_objtype\_industrial\_area  
 SOSI Elements;16  
 sosi\_objtype\_lake  
 SOSI Elements;16  
 sosi\_objtype\_lake\_edge  
 SOSI Elements;16  
 sosi\_objtype\_lake\_river\_barrier  
 SOSI Elements;16  
 sosi\_objtype\_land\_use\_boundary

SOSI Elements;16  
 sosi\_objtype\_lane  
 SOSI Elements;16  
 sosi\_objtype\_level\_crossing  
 SOSI Elements;16  
 sosi\_objtype\_marsh  
 SOSI Elements;16  
 sosi\_objtype\_municipal\_divide  
 SOSI Elements;16  
 sosi\_objtype\_municipality  
 SOSI Elements;16  
 sosi\_objtype\_municipality\_boundary  
 SOSI Elements;16  
 sosi\_objtype\_national\_border  
 SOSI Elements;16  
 sosi\_objtype\_open\_land  
 SOSI Elements;16  
 sosi\_objtype\_pedestrian\_bicycle\_road\_centre\_line  
 SOSI Elements;16  
 sosi\_objtype\_river\_brook  
 SOSI Elements;16  
 sosi\_objtype\_river\_brook\_edge  
 SOSI Elements;16  
 sosi\_objtype\_road\_block  
 SOSI Elements;16  
 sosi\_objtype\_road\_centre\_line  
 SOSI Elements;16  
 sosi\_objtype\_road\_under\_railway  
 SOSI Elements;16  
 sosi\_objtype\_sea\_river\_delineation  
 SOSI Elements;16  
 sosi\_objtype\_sea\_surface  
 SOSI Elements;16  
 sosi\_objtype\_sidewalk  
 SOSI Elements;16  
 sosi\_objtype\_snow\_field  
 SOSI Elements;16  
 sosi\_objtype\_spelling  
 SOSI Elements;16  
 sosi\_objtype\_stone\_quarry  
 SOSI Elements;16  
 sosi\_objtype\_street\_address  
 SOSI Elements;16  
 sosi\_objtype\_territorial\_boundary  
 SOSI Elements;16  
 sosi\_objtype\_turn\_connecting\_segment  
 SOSI Elements;16  
 sosi\_objtype\_unknown  
 SOSI Elements;15  
 SosiCharsetSingleton  
 sosicon::sosi::SosiCharsetSingleton;145  
 SosiChildrenIterator  
 SOSI Elements;13  
 SosiChildrenList  
 SOSI Elements;13  
 sosicon;18

- byte\_order.cpp;174
- CoordinateList;19
- flush;20
- getNext;20
- getNextOffset;20
- isClockwise;20
- isCounterClockwise;20
- logstream;21
- neListToCoordList;21
- Wkt;19
- wkt\_linestring;20
- wkt\_point;19
- wkt\_polygon;20
- wkt\_unknown;19
- sosicon::byteOrder;22
  - big;22
  - determine;22
  - doubleToLittleEndian;23
  - endianness;24
  - Endianness;22
  - little;22
  - not\_set;22
  - toBigEndian;23
  - toLittleEndian;23
- sosicon::CommandLine;35
  - ~CommandLine;36
  - CommandLine;36
  - mAppend;38
  - mCommand;38
  - mCreateStatements;38
  - mDbSchema;38
  - mDbTable;38
  - mDestinationDirectory;38
  - mFieldSelection;39
  - mFilterSosiId;39
  - mGeomTypes;39
  - mIncludeHeader;39
  - mInsertStatements;39
  - mIsTtyIn;39
  - mIsTtyOut;40
  - mMakeSubDir;40
  - mObjTypes;40
  - mOutputFile;40
  - mSourceFiles;40
  - mSrid;40
  - mVerbose;41
  - outputDisclaimer;37
  - outputHelpText;37
  - outputLicense;37
  - parse;37
- sosicon::ConverterSosi2psql;42
  - ~ConverterSosi2psql;44
  - buildCreateStatement;44
  - buildCreateStatements;44
  - buildInsertStatement;45
  - buildInsertStatements;45
  - cleanup;45; 46
  - ConverterSosi2psql;44
  - extractData;46
  - FieldsList;43
  - FieldsListCollection;43
  - getSrid;46
  - init;46
  - insertLineString;47
  - insertPoint;47
  - insertPolygon;47
  - makePsql;48
  - mCmd;49
  - mCurrentSourcefile;49
  - mFieldsListCollection;49
  - mRowsListCollection;49
  - objTypeExcluded;48
  - RowsList;43
  - RowsListCollection;44
  - run;48
  - writePsql;49
- sosicon::ConverterSosi2psql::Field;78
  - expand;78
  - Field;78
  - isNumeric;78
  - length;78
  - mIsNumeric;79
  - mMaxLength;79
  - mMinLength;79
- sosicon::ConverterSosi2shp;50
  - ~ConverterSosi2shp;51
  - ConverterSosi2shp;51
  - init;51
  - makeBasePath;51
  - makeShp;51
  - mCmd;52
  - mCurrentSourcefile;52
  - run;51
  - writeFile;52
- sosicon::ConverterSosi2tsv;54
  - ~ConverterSosi2tsv;54
  - ConverterSosi2tsv;54
  - init;55
  - mCmd;55
  - run;55
- sosicon::ConverterSosi2xml;56
  - ~ConverterSosi2xml;57
  - ConverterSosi2xml;56
  - init;57
  - makeXML;57
  - mCmd;57
  - run;57
- sosicon::ConverterSosiStat;58
  - ~ConverterSosiStat;59
  - ConverterSosiStat;59
  - init;59
  - makeStat;59

- mCmd;60
- mGeoTypes;60
- mObjTypes;60
- printElementData;59
- printListContent;59
- printTableHeader;60
- run;60
- sosicon::Coordinate;61
  - ~Coordinate;62
  - Coordinate;62
  - divide;62
  - equals;62
  - getE;62
  - getN;62
  - leftOf;63
  - mAltitude;64
  - mEast;64
  - mNorth;64
  - rightOf;63
  - setE;63
  - setH;63
  - setN;63
  - shift;63
  - toString;63
- sosicon::CoordinateCollection;65
  - ~CoordinateCollection;66
  - CoordinateCollection;66
  - discoverCoords;66
  - extractPath;66
  - free;67
  - getGeom;67
  - getGeomSizes;67
  - getHoles;67
  - getHoleSizes;67
  - getNextInGeom;67
  - getNumPartsGeom;67
  - getNumPartsHoles;67
  - getNumPointsGeom;68
  - getNumPointsHoles;68
  - getXmax;68
  - getXmin;68
  - getYmax;68
  - getYmin;68
  - mGeom;68
  - mGeomIndex;68
  - mGeomNormalized;68
  - mGeomSizes;68
  - mHoles;69
  - mHoleSizes;69
  - mHolesNormalized;69
  - mNumPartsGeom;69
  - mNumPartsHoles;69
  - mNumPointsGeom;69
  - mNumPointsHoles;69
  - mXmax;69
  - mXmin;69

- mYmax;69
- mYmin;69
- sosicon::EventDispatcher
  - addEventListener;75
  - Dispatch;75
  - ListenerLst;74
  - mListeners;75
  - removeEventListener;75
- sosicon::EventDispatcher::Listener
  - ~Listener;118
  - onEvent;118
- sosicon::EventDispatcher< Event >;74
- sosicon::EventDispatcher< Event >::Listener;118
- sosicon::Factory;76
  - get;76
  - release;76
- sosicon::IBinaryStreamable;80
  - ~IBinaryStreamable;80
  - writeBinary;80
- sosicon::IConverter;82
  - ~IConverter;82
  - init;82
  - run;83
- sosicon::ICoordinate;84
  - ~ICoordinate;85
  - divide;85
  - equals;85
  - getE;85
  - getN;85
  - leftOf;85
  - rightOf;85
  - setE;86
  - setH;86
  - setN;86
  - shift;86
  - toString;86
- sosicon::ILookupTable;87
  - ~ILookupTable;87
  - get;87
  - toString;87
- sosicon::IRectangle;94
  - ~IRectangle;94
  - bottom;95
  - left;95
  - right;95
  - top;96
- sosicon::IShapeElement;97
  - ~IShapeElement;97
  - getByteSize;98
  - getMBR;98
  - getSosiElement;98
  - getWordSize;98
  - populate;98
- sosicon::IShapeElementHeader;100
  - ~IShapeElementHeader;100
- sosicon::IShapefile;101

- ~IShapefile;101
- build;102
- filterSosiId;102
- sosicon::IShapefileDbfPart;103
  - writeBinary;103
  - writeDbf;104
- sosicon::IShapefilePrjPart;105
  - writeBinary;105
  - writePrj;106
- sosicon::IShapefileShpPart;107
  - writeBinary;107
  - writeShp;108
- sosicon::IShapefileShxPart;109
  - writeBinary;109
  - writeShx;110
- sosicon::IShapeHeader;111
  - ~IShapeHeader;111
  - getBoundingBox;112
  - getByteSize;112
  - getFileLength;112
  - getShapeType;112
  - getWordSize;112
  - setBoundingBox;112
  - setFileLength;112
  - setShapeType;112
- sosicon::ISosiElement;113
  - ~ISosiElement;114
  - addChild;114
  - children;114
  - deleteChildren;114
  - dump;114
  - find;114
  - getChild;114
  - getData;114
  - getLevel;114
  - getName;114
  - getObjType;115
  - getRoot;115
  - getSerial;115
  - getType;115
- sosicon::ISosiHeadMember;116
  - ~ISosiHeadMember;116
  - init;116
  - initialized;116
- sosicon::LogEvent;119
  - LogEvent;119
  - mMessage;119
  - mUpdate;119
- sosicon::LogEventDispatcher;121
- sosicon::Logger;122
  - addEventListener;122
  - mLogEventDispatcher;123
  - mMsgStream;123
  - operator<<;122; 123
  - removeEventListener;123
- sosicon::Parser;124

- ~Parser;125
- complete;125
- digestPendingElement;125
- dump;126
- getRootElement;126
- mCurrentCharset;126
- mElementIndex;126
- mElementStack;126
- mPendingElementAttributes;127
- mPendingElementLevel;127
- mPendingElementName;127
- mPendingElementSerial;127
- Parser;125
- ragelParseSosiLine;126
- sosicon::shape;25
  - DbfFieldLengths;25
  - DbfRecord;25
  - DbfRecordSet;25
  - getShapeEquivalent;26
  - shape\_type\_multiPatch;26
  - shape\_type\_multipoint;26
  - shape\_type\_multiPointM;26
  - shape\_type\_multipointZ;26
  - shape\_type\_none;26
  - shape\_type\_nullShape;26
  - shape\_type\_point;26
  - shape\_type\_pointM;26
  - shape\_type\_pointZ;26
  - shape\_type\_polygon;26
  - shape\_type\_polygonM;26
  - shape\_type\_polygonZ;26
  - shape\_type\_polyLine;26
  - shape\_type\_polyLineM;26
  - shape\_type\_polyLineZ;26
  - ShapeType;26
  - ShxOffsets;26
- sosicon::shape::DoubleField;73
  - b;73
  - d;73
- sosicon::shape::Int16Field;90
  - b;90
  - i;90
- sosicon::shape::Int32Field;91
  - b;91
  - i;91
- sosicon::shape::Int32TField;92
  - b;92
  - i;92
  - t;92
- sosicon::shape::Int8Field;93
  - b;93
  - i;93
- sosicon::shape::Shapefile;129
  - ~Shapefile;132
  - adjustMasterMbr;132
  - build;132

- buildDbf;132
- buildDbfFieldDescriptor;133
- buildDbfHeader;133
- buildDbfRecordSection;133
- buildShpElement;134
- buildShpHeader;134
- buildShpPoint;134
- buildShpPolygon;134
- buildShpPolyLine;135
- buildShpRecCoordinate;135
- buildShpRecCoordinates;135
- buildShpRecHeaderCommonPart;136
- buildShpRecHeaderExtended;136
- buildShpRecHeaderOffsets;136
- buildShx;137
- expandShpBuffer;137
- extractDbfFields;137
- filterSosiId;137
- getNormalized;138
- insertDbfRecord;138
- insertShxOffset;138
- MAX\_BUFFER\_CHUNK\_SIZE;139
- mDbfBuffer;139
- mDbfBufferSize;139
- mDbfFieldLengths;140
- mDbfHeader;140
- mDbfRecordSet;140
- mFilterSosiId;140
- mFilterSosiObjTypes;140
- mRecordNumber;140
- mShpBuffer;140
- mShpBufferSize;140
- mShpHeader;140
- mShpSize;141
- mShxBuffer;141
- mShxBufferSize;141
- mShxHeader;141
- mShxOffsets;141
- mSosiTree;141
- mXmax;141
- mXmin;141
- mYmax;142
- mYmin;142
- saveToDbf;138
- Shapefile;132
- writeDbf;138
- writePrj;139
- writeShp;139
- writeShx;139
- sosicon::shape::ShxIndex;143
  - length;143
  - offset;143
- sosicon::sosi;27
  - deleteNorthEast;28
  - NorthEastList;28
- sosicon::sosi::chartables;29
- sosicon::sosi::CoordSys;71
  - CoordSys;71
  - displayString;71
  - mDisplayString;72
  - mPrjString;72
  - mSrid;72
  - mSysCode;72
  - prjString;72
  - srid;72
  - valid;72
- sosicon::sosi::ReferenceData;128
  - reverse;128
  - serial;128
  - subtract;128
- sosicon::sosi::SosiCharsetSingleton;144
  - ~SosiCharsetSingleton;145
  - getEncoding;145
  - getEncodingName;145
  - getInstance;145
  - init;145
  - initialized;146
  - mCharset;146
  - mCharsetName;146
  - mInitialized;146
  - mInstance;146
  - mSosiElement;146
  - SosiCharsetSingleton;145
  - toIso8859\_1;146
  - utf8ToIso8859\_1;146
- sosicon::sosi::SosiElement;148
  - addChild;149
  - children;150
  - deleteChildren;150
  - dump;150
  - find;150
  - getChild;150
  - getData;150
  - getLevel;150
  - getName;151
  - getObjType;151
  - getRoot;151
  - getSerial;151
  - getType;151
  - mChildren;152
  - mData;152
  - mIndex;152
  - mLevel;152
  - mName;152
  - mObjType;152
  - mObjTypeStr;152
  - mRoot;152
  - mSerial;153
  - mTranslation;153
  - mType;153
  - nextChild;151
  - SosiElement;149



- sosicon::sosi::SosiElementSearch;154
  - element;155
  - index;155
  - matchTypes;155
  - mElementTypes;156
  - mIndex;156
  - mSosiElement;156
  - next;155
  - SosiElementSearch;154
  - type;155
  - types;155
- sosicon::sosi::SosiJunctionPoint;157
  - ~SosiJunctionPoint;157
  - mSosiElement;157
  - SosiJunctionPoint;157
- sosicon::sosi::SosiNorthEast;159
  - ~SosiNorthEast;160
  - append;160
  - back;160
  - dump;161
  - expandBoundingBox;161
  - free;161
  - front;161
  - getNext;161
  - getNumPoints;161
  - initHeadMember;161
  - mCoordinates;162
  - mCoordinatesIterator;162
  - mMaxX;162
  - mMaxY;162
  - mMinX;162
  - mMinY;162
  - mOrigo;162
  - mSosiElement;162
  - mUnit;163
  - operator/=;161
  - operator+=;161
  - ragelParseCoordinatesNe;161
  - ragelParseCoordinatesNeh;162
  - reverse;162
  - SosiNorthEast;160
- sosicon::sosi::SosiOrigoNE;164
  - ~SosiOrigoNE;165
  - getE;165
  - getN;165
  - init;165
  - initialized;165
  - mInitialized;165
  - mOrigoE;166
  - mOrigoN;166
  - mSosiElement;166
  - ragelParseSosiOrigoNE;165
  - SosiOrigoNE;164; 165
- sosicon::sosi::SosiRefList;167
  - ~SosiRefList;168
  - getNextGeometry;168

- mRefListCollection;168
- mRefListCollectionIndex;168
- mRefListIndex;168
- mSosiElement;168
- ragelParseSosiRef;168
- SosiRefList;167
- sosicon::sosi::SosiTranslationTable;169
  - MAX\_COORDSYS\_TABLE;170
  - mCoordSysTable;170
  - mObjTypeNameMap;171
  - mSosiCharset;171
  - mTypeNameMap;171
  - reverseLookup;170
  - sosiNameToType;170
  - sosiObjNameToType;170
  - SosiTranslationTable;169
  - sosiTypeToName;170
  - sosiTypeToObjName;170
  - sysCodeToCoordSys;170
- sosicon::sosi::SosiUnit;172
  - ~SosiUnit;172
  - getDivisor;173
  - init;173
  - initialized;173
  - mDivisor;173
  - mInitialized;173
  - mSosiElement;173
  - SosiUnit;172; 173
- sosicon::utils;30
  - className2FileName;31
  - explode;31
  - fileExists;31
  - getPathInfo;31
  - isNumeric;31
  - nonExistingFilename;32
  - normalizeAppClassName;32
  - purgeCrLf;32
  - repeat;32
  - replaceAll;33
  - sqlNormalize;33
  - stripTrailingSlash;33
  - toFieldname;33
  - toLower;33
  - trim;33
  - trimLeft;34
  - trimRight;34
  - ucFirst;34
  - unquote;34
  - wktToStr;34
- SosiElement
  - sosicon::sosi::SosiElement;149
- SosiElementMap
  - SOSI Elements;13
- SosiElementSearch
  - sosicon::sosi::SosiElementSearch;154
- SosiJunctionPoint

- sosicon::sosi::SosiJunctionPoint;157
- sosiNameToType
  - SOSI Elements;17
  - sosicon::sosi::SosiTranslationTable;170
- SosiNorthEast
  - sosicon::sosi::SosiNorthEast;160
- sosiObjNameToType
  - SOSI Elements;17
  - sosicon::sosi::SosiTranslationTable;170
- SosiOrigoNE
  - sosicon::sosi::SosiOrigoNE;164; 165
- SosiRefList
  - sosicon::sosi::SosiRefList;167
- SosiTranslationTable
  - sosicon::sosi::SosiTranslationTable;169
- sosiTypeToName
  - sosicon::sosi::SosiTranslationTable;170
- sosiTypeToObjName
  - sosicon::sosi::SosiTranslationTable;170
- SosiUnit
  - sosicon::sosi::SosiUnit;172; 173
- sqlNormalize
  - sosicon::utils;33
- srid
  - sosicon::sosi::CoordSys;72
- stripTrailingSlash
  - sosicon::utils;33
- strtoimax
  - inttypes.h;229
- strtoumax
  - inttypes.h;229
- subtract
  - sosicon::sosi::ReferenceData;128
- sysCodeToCoordSys
  - SOSI Elements;17
  - sosicon::sosi::SosiTranslationTable;170
- t
  - sosicon::shape::Int32TField;92
- toBigEndian
  - sosicon::byteOrder;23
- toFieldname
  - sosicon::utils;33
- toIso8859\_1
  - sosicon::sosi::SosiCharsetSingleton;146
- toLittleEndian
  - sosicon::byteOrder;23
- toLower
  - sosicon::utils;33
- top
  - sosicon::IRectangle;96
- toString
  - sosicon::Coordinate;63
  - sosicon::ICoordinate;86
  - sosicon::ILookupTable;87
- trim

- sosicon::utils;33
- trimLeft
  - sosicon::utils;34
- trimRight
  - sosicon::utils;34
- type
  - sosicon::sosi::SosiElementSearch;155
- types
  - sosicon::sosi::SosiElementSearch;155
- ucFirst
  - sosicon::utils;34
- unquote
  - sosicon::utils;34
- utf8ToIso8859\_1
  - sosicon::sosi::SosiCharsetSingleton;146
- valid
  - sosicon::sosi::CoordSys;72
- wcstoimax
  - inttypes.h;229
- wcstoumax
  - inttypes.h;229
- Wkt
  - sosicon;19
- wkt\_linestring
  - sosicon;20
- wkt\_point
  - sosicon;19
- wkt\_polygon
  - sosicon;20
- wkt\_unknown
  - sosicon;19
- wktToStr
  - sosicon::utils;34
- writeBinary
  - sosicon::IBinaryStreamable;80
  - sosicon::IShapefileDbfPart;103
  - sosicon::IShapefilePrjPart;105
  - sosicon::IShapefileShpPart;107
  - sosicon::IShapefileShxPart;109
- writeDbf
  - sosicon::IShapefileDbfPart;104
  - sosicon::shape::Shapefile;138
- writeFile
  - sosicon::ConverterSosi2shp;52
- writePrj
  - sosicon::IShapefilePrjPart;106
  - sosicon::shape::Shapefile;139
- writePsql
  - sosicon::ConverterSosi2psql;49
- writeShp
  - sosicon::IShapefileShpPart;108
  - sosicon::shape::Shapefile;139
- writeShx
  - sosicon::IShapefileShxPart;110
  - sosicon::shape::Shapefile;139