sosicon

Version v 0.1 prerelease
30.09.12 10.01

Table of Contents

Module Index 2
Namespace Index 3
Hierarchical Index 4
Class Index 5
File Index........ 6
Module Documentation 7
- GIS entities 7
- Converters 7
- Interfaces 7
- SOSI Elements 8
- sosicon 9
- sosicon::stringUtils 10

Class Documentation........ 13
- sosicon::AddressUnit........ 13
- sosicon::CadastralUnit........ 16
- sosicon::CommandLine 19
- sosicon::ConverterSosi2shp 23
- sosicon::ConverterSosi2tsv 25
- sosicon::CoordSys 27
- sosicon::Factory........ 31
- sosicon::IConverter 33
- sosicon::ISosiElement 35
- sosicon::Parser........ 39
- sosicon::SosiElement........ 46
- sosicon::SosiElementHead 50
- sosicon::SosiElementPoint 54
- sosicon::SosiElementText 60

File Documentation 64
- /prosjekter/sosicon/src/address_unit.cpp 64
- /prosjekter/sosicon/src/address_unit.h........ 65
- /prosjekter/sosicon/src/cadastral_unit.cpp........ 66
- /prosjekter/sosicon/src/cadastral_unit.h 67
- /prosjekter/sosicon/src/command_line.cpp 68
- /prosjekter/sosicon/src/command_line.h 69
- /prosjekter/sosicon/src/converter_sosi2shp.cpp 70
- /prosjekter/sosicon/src/converter_sosi2shp.h 71
- /prosjekter/sosicon/src/converter_sosi2tsv.cpp 72
- /prosjekter/sosicon/src/converter_sosi2tsv.h 73
- /prosjekter/sosicon/src/coord_sys.cpp 74
- /prosjekter/sosicon/src/coord_sys.h........ 75
- /prosjekter/sosicon/src/factory.cpp........ 76
- /prosjekter/sosicon/src/factory.h........ 77
- /prosjekter/sosicon/src/i_converter.h........ 78
- /prosjekter/sosicon/src/i_sosi_element.h 79
- /prosjekter/sosicon/src/main.cpp 80
- /prosjekter/sosicon/src/main.h........ 81
- /prosjekter/sosicon/src/parser.cpp 82
- /prosjekter/sosicon/src/parser.h 83
- /prosjekter/sosicon/src/parser/parser_sosi_line.rl........ 84
- /prosjekter/sosicon/src/parser_pathinfo.cpp 85
- /prosjekter/sosicon/src/parser_sosi_line.cpp 86
- /prosjekter/sosicon/src/sosi_element.cpp 87

/prosjekter/sosicon/src/sosi_element.h 88
/prosjekter/sosicon/src/sosi_element_head.h 89
/prosjekter/sosicon/src/sosi_element_point.cpp 90
/prosjekter/sosicon/src/sosi_element_point.h 91
/prosjekter/sosicon/src/sosi_element_text.h 92
/prosjekter/sosicon/src/string_utils.cpp 93
/prosjekter/sosicon/src/string_utils.h 94
Index 95

Module Index

Modules

Here is a list of all modules:

GIS entities 7
Converters 7
Interfaces 7
SOSI Elements 8

Namespace Index

Namespace List

Here is a list of all namespaces with brief descriptions:

sosicon (Application root) 9
sosicon::stringUtils (String manipulation routines) 10

Hierarchical Index

Class Hierarchy

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

- sosicon::AddressUnit 13
- sosicon::CadastralUnit 16
- sosicon::CommandLine 19
- sosicon::CoordSys 27
- sosicon::Factory 31
- sosicon::IConverter 33
  - sosicon::ConverterSosi2shp 23
  - sosicon::ConverterSosi2tsv 25
- sosicon::ISosiElement 35
  - sosicon::SosiElement 46
  - sosicon::SosiElementHead 50
  - sosicon::SosiElementPoint 54
  - sosicon::SosiElementText 60
- sosicon::Parser 39

Class Index

Class List

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

sosicon::AddressUnit (Address usage unit) 13
sosicon::CadastralUnit (Cadastral unit) 16
sosicon::CommandLine (Command-line parser) 19
sosicon::ConverterSosi2shp (SOSI to ESRI Shape converter) 23
sosicon::ConverterSosi2tsv (SOSI to TSV converter) 25
sosicon::CoordSys (Coordinate system) 27
sosicon::Factory (Factory class) 31
sosicon::IConverter (Interface: Converter) 33
sosicon::ISosiElement (Interface: SOSI element) 35
sosicon::Parser (SOSI file parser) 39
sosicon::SosiElement (Basic SOSI element) 46
sosicon::SosiElementHead (Sosi head element (HODE)) 50
sosicon::SosiElementPoint (Sosi point element (PUNKT)) 54
sosicon::SosiElementText (Sosi text element (TEKST)) 60

File Index

File List

Here is a list of all files with brief descriptions:

/prosjekter/sosicon/src/address_unit.cpp 64
/prosjekter/sosicon/src/address_unit.h 65
/prosjekter/sosicon/src/cadastral_unit.cpp 66
/prosjekter/sosicon/src/cadastral_unit.h 67
/prosjekter/sosicon/src/command_line.cpp 68
/prosjekter/sosicon/src/command_line.h 69
/prosjekter/sosicon/src/converter_sosi2shp.cpp 70
/prosjekter/sosicon/src/converter_sosi2shp.h 71
/prosjekter/sosicon/src/converter_sosi2tsv.cpp 72
/prosjekter/sosicon/src/converter_sosi2tsv.h 73
/prosjekter/sosicon/src/coord_sys.cpp 74
/prosjekter/sosicon/src/coord_sys.h 75
/prosjekter/sosicon/src/factory.cpp 76
/prosjekter/sosicon/src/factory.h 77
/prosjekter/sosicon/src/i_converter.h 78
/prosjekter/sosicon/src/i_sosi_element.h 79
/prosjekter/sosicon/src/main.cpp 80
/prosjekter/sosicon/src/main.h 81
/prosjekter/sosicon/src/parser.cpp 82
/prosjekter/sosicon/src/parser.h 83
/prosjekter/sosicon/src/parser_pathinfo.cpp 85
/prosjekter/sosicon/src/parser_sosi_line.cpp 86
/prosjekter/sosicon/src/sosi_element.cpp 87
/prosjekter/sosicon/src/sosi_element.h 88
/prosjekter/sosicon/src/sosi_element_head.h 89
/prosjekter/sosicon/src/sosi_element_point.cpp 90
/prosjekter/sosicon/src/sosi_element_point.h 91
/prosjekter/sosicon/src/sosi_element_text.h 92
/prosjekter/sosicon/src/string_utils.cpp 93
/prosjekter/sosicon/src/string_utils.h 94
/prosjekter/sosicon/src/parser/parser_sosi_line.rl 84

Module Documentation

GIS entities

Classes

- class **sosicon::AddressUnit**
- *Address usage unit.* class **sosicon::CadastralUnit**
- *Cadastral unit.* class **sosicon::CoordSys**

Coordinate system.

Detailed Description

Collection of objects representing different aspects of geographical features embedded in the SOSI file format.

Converters

Classes

- class **sosicon::ConverterSosi2shp**
- *SOSI to ESRI Shape converter.* class **sosicon::ConverterSosi2tsv**

SOSI to TSV 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:

- -2tsv: **sosicon::ConverterSosi2tsv** TSV file conversion
- -2shp: **sosicon::ConverterSosi2shp** Shape file conversion

Interfaces

Classes

- class **sosicon::IConverter**
- *Interface: Converter.* class **sosicon::ISosiElement**

Interface: SOSI element.

Detailed Description

This is a listing of generic interfaces used within sosicon.

SOSI Elements

Classes

- class **sosicon::SosiElement**
- *Basic SOSI element.* class **sosicon::SosiElementHead**
- *Sosi head element (HODE)* class **sosicon::SosiElementPoint**
- *Sosi point element (PUNKT)* class **sosicon::SosiElementText**

Sosi text element (TEKST)

Detailed Description

Implemented representation of SOSI file elements.

Namespace Documentation

sosicon Namespace Reference

Application root.

Namespaces

- namespace **stringUtils**

String manipulation routines. Classes

- class **AddressUnit**
- *Address usage unit.* class **CadastralUnit**
- *Cadastral unit.* class **CommandLine**
- *Command-line parser.* class **ConverterSosi2shp**
- *SOSI to ESRI Shape converter.* class **ConverterSosi2tsv**
- *SOSI to TSV converter.* class **CoordSys**
- *Coordinate system.* class **Factory**
- ***Factory** class.* class **IConverter**
- *Interface: Converter.* class **ISosiElement**
- *Interface: SOSI element.* class **Parser**
- *SOSI file parser.* class **SosiElement**
- *Basic SOSI element.* class **SosiElementHead**
- *Sosi head element (HODE)* class **SosiElementPoint**
- *Sosi point element (PUNKT)* class **SosiElementText**

Sosi text element (TEKST)

Detailed Description

Application root.

sosicon::stringUtils Namespace Reference

String manipulation routines.

Functions

- std::string **className2FileName** (const std::string &className)
  Converts Class name to file name string.
- std::string **normalizeAppClassName** (const std::string &className)
  Asserts correct name of application classes.
- 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 occurences of one string witn another.
- 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 **toLower** (const std::string &from)
- std::string **ucFirst** (const std::string &str)
- void **getPathInfo** (std::string path, std::string &dir, std::string &tit, std::string &ext)

Detailed Description

String manipulation routines.

Function Documentation

string sosicon::stringUtils::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:

className	The pascal-cased class name to convert to a file name.

Returns:

The file name string without extension.

Definition at line 23 of file string_utils.cpp.

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

Definition at line 146 of file string_utils.cpp.

string sosicon::stringUtils::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:

className	The class name string to be resolved and normalized.

Returns:

Normalized and corrected class name string.

Definition at line 44 of file string_utils.cpp.

string sosicon::stringUtils::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:

seq	Reference to the string to be repeated.
count	The numner of times to repeat the string sequence.

Returns:

The result string.

Definition at line 64 of file string_utils.cpp.

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

Replace all occurences of one string witn another.

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

Parameters:

from	The string sequence to be changed.
to	The string to replace the 'from' sequence with.
subject	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 75 of file string_utils.cpp.

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

Definition at line 89 of file string_utils.cpp.

string sosicon::stringUtils::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:

str	The target string.

Returns:

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

Definition at line 105 of file string_utils.cpp.

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

Definition at line 111 of file string_utils.cpp.

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

Definition at line 118 of file string_utils.cpp.

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

Definition at line 126 of file string_utils.cpp.

Class Documentation

sosicon::AddressUnit Class Reference

Address usage unit.

```
#include <address_unit.h>
```

Public Member Functions

- **AddressUnit** ()
  Constructor.
- virtual **~AddressUnit** ()
  Destructor.
- std::string **toString** ()

Public Attributes

- std::string **mFloorLevel**
  SOSI Address Unit field.
- std::string **mFloorNumber**
  SOSI Address Unit field.
- std::string **mDoorNumber**
  SOSI Address Unit field.

Detailed Description

Address usage unit.

Author:

Espen Andersen

Copyright:

Represents address usage number. Since an address may consist of several housing units, this entitiy identify individual units (apartments or sections) on a given address. The address unit number consists of the following fields:

- Floor level
- Floor number
- Door number

Definition at line 43 of file address_unit.h.

Constructor & Destructor Documentation

sosicon::AddressUnit::AddressUnit ()

Constructor.

Definition at line 21 of file address_unit.cpp.

sosicon::AddressUnit::~AddressUnit ()`[virtual]`

Destructor.

Definition at line 24 of file address_unit.cpp.

Member Function Documentation

std::string sosicon::AddressUnit::toString ()

Definition at line 27 of file address_unit.cpp.

Member Data Documentation

std::string sosicon::AddressUnit::mDoorNumber

SOSI Address Unit field.

Indicating the door number for current address unit. The number sequence starts at 01 and increments by one for each door, counting left-to-right.

Definition at line 76 of file address_unit.h.

std::string sosicon::AddressUnit::mFloorLevel

SOSI Address Unit field.

Indicating at what floor level current address unit has its main entry. May be one of the following values:

- "K" = Basement
- "U" = Below ground level
- "H" = Over ground level
- "L" = Attic

Definition at line 56 of file address_unit.h.

std::string sosicon::AddressUnit::mFloorNumber

SOSI Address Unit field.

Indicating the floor number for current address unit. The number sequence starts at 01 for each floor level.

For floor levels over ground level, the lowermost floor is assigned to 01 and higher floors get higher numbers.

For floor levels below ground level, the uppermost floor is assigned to 01 and lower floors get higher numbers.

Definition at line 69 of file address_unit.h.

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

- /prosjekter/sosicon/src/**address_unit.h**
- /prosjekter/sosicon/src/**address_unit.cpp**

sosicon::CadastralUnit Class Reference

Cadastral unit.

```
#include <cadastral_unit.h>
```

Public Member Functions

- **CadastralUnit** ()
  Constructor.
- virtual **~CadastralUnit** ()
  Destructor.
- std::string **toString** ()
  Return string representation.

Public Attributes

- std::string **mMunicipality**
  Municipality number.
- std::string **mCadastre**
  Cadastral unit number.
- std::string **mProperty**
  Property number.
- std::string **mLeasehold**
  Leasehold number.
- std::string **mSection**
  Section number.

Detailed Description

Cadastral unit.

Author:

Espen Andersen

Copyright:

GNU General Public License

Represents a property as registered in the national cadastre. Some properties do not have street addresses. Instead, they are identified by their cadastral unit code consisting of the following parts:

- Municipality number
- Cadastral unit number
- Property number
- Section number
- Leashold number

Definition at line 42 of file cadastral_unit.h.

Constructor & Destructor Documentation

sosicon::CadastralUnit::CadastralUnit ()

Constructor.

Definition at line 21 of file cadastral_unit.cpp.

sosicon::CadastralUnit::~CadastralUnit () `[virtual]`

Destructor.

Definition at line 24 of file cadastral_unit.cpp.

Member Function Documentation

std::string sosicon::CadastralUnit::toString ()

Return string representation.

Creates a human-readable string representation for this cadastral unit number.

Returns:

string representation of current object.

Definition at line 27 of file cadastral_unit.cpp.

Member Data Documentation

std::string sosicon::CadastralUnit::mCadastre

Cadastral unit number.

Definition at line 50 of file cadastral_unit.h.

std::string sosicon::CadastralUnit::mLeasehold

Leasehold number.

Definition at line 56 of file cadastral_unit.h.

std::string sosicon::CadastralUnit::mMunicipality

Municipality number.

Definition at line 47 of file cadastral_unit.h.

std::string sosicon::CadastralUnit::mProperty

Property number.

Definition at line 53 of file cadastral_unit.h.

std::string sosicon::CadastralUnit::mSection

Section number.

Definition at line 59 of file cadastral_unit.h.

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

- /prosjekter/sosicon/src/**cadastral_unit.h**
- /prosjekter/sosicon/src/**cadastral_unit.cpp**

sosicon::CommandLine Class Reference

Command-line parser.

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

Public Member Functions

- void **outputHelpText** ()
  Display help text.
- void **parse** (int argc, char *argv[])
  Read command-line arguments.
- **CommandLine** ()
  Constructor.
- virtual **~CommandLine** ()
  Destructor.

Public Attributes

- std::string **mCommand**
  Conversion command.
- std::vector< std::string > **mSourceFiles**
  List of input files.
- std::vector< std::string > **mObjTypes**
  List of object types to output.
- std::vector< std::string > **mFieldSelection**
  List of selected fields.
- std::string **mDestinationDirectory**
  Destination directory.
- std::string **mOutputFile**
  Destination file.
- bool **mAppend**
  Append flag.
- bool **mIncludeHeader**
  Include column headers.
- 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 37 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 27 of file command_line.cpp.

Member Function Documentation

void sosicon::CommandLine::outputHelpText ()

Display help text.

Outputs simple help text to the command-line.

Definition at line 119 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:

argc	Number of arguments present. Passed on from **main()** function.
argv	Array of string pointers to each argument. Passed on from **main()** function.

Definition at line 30 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 91 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 46 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 77 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 70 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 98 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 63 of file command_line.h.

std::string sosicon::CommandLine::mOutputFile

Destination file.

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

Definition at line 83 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 55 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 i 2 (-V), a more comprehensive summary of every SOSI element in all source files will be output.

Definition at line 107 of file command_line.h.

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

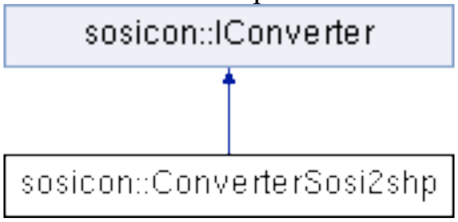
- /prosjekter/sosicon/src/**command_line.h**
- /prosjekter/sosicon/src/**command_line.cpp**

sosicon::ConverterSosi2shp Class Reference

SOSI to ESRI Shape converter.

`#include <converter_sosi2shp.h>`

Inheritance diagram for sosicon::ConverterSosi2shp:



Public Member Functions

- **ConverterSosi2shp** ()
  Constructor.
- virtual void **init** (**CommandLine** cmd)
  Initialize converter.
- virtual void **run** ()
  Start conversion.

Private Member Functions

- virtual **~ConverterSosi2shp** ()

Private Attributes

- **CommandLine mCmd**

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 46 of file converter_sosi2shp.h.

Constructor & Destructor Documentation

sosicon::ConverterSosi2shp::~ConverterSosi2shp () `[private]`, `[virtual]`

Definition at line 26 of file converter_sosi2shp.cpp.

sosicon::ConverterSosi2shp::ConverterSosi2shp ()

Constructor.

Definition at line 21 of file converter_sosi2shp.cpp.

Member Function Documentation

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

Initialize converter.

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

See Also:

sosicon::IConverter::init()

Implements **sosicon::IConverter** (*p.34*).

Definition at line 31 of file converter_sosi2shp.cpp.

void sosicon::ConverterSosi2shp::run ()`[virtual]`

Start conversion.

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

See Also:

sosicon::IConverter::run()

Implements **sosicon::IConverter** (*p.34*).

Definition at line 36 of file converter_sosi2shp.cpp.

Member Data Documentation

CommandLine sosicon::ConverterSosi2shp::mCmd`[private]`

Definition at line 48 of file converter_sosi2shp.h.

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

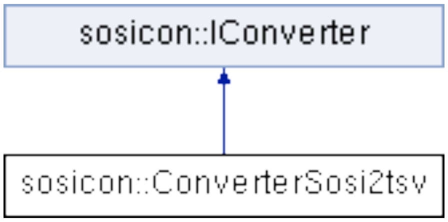
- /prosjekter/sosicon/src/**converter_sosi2shp.h**
- /prosjekter/sosicon/src/**converter_sosi2shp.cpp**

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** ()
  Start conversion.

Private Member Functions

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

Private Attributes

- **CommandLine mCmd**

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

sosicon::ConverterSosi2tsv::~ConverterSosi2tsv () `[private]`, `[virtual]`

Destructor.

Definition at line 26 of file converter_sosi2tsv.cpp.

sosicon::ConverterSosi2tsv::ConverterSosi2tsv ()

Constructor.

Definition at line 21 of file converter_sosi2tsv.cpp.

Member Function Documentation

void sosicon::ConverterSosi2tsv::init (sosicon::CommandLine *cmd*)`[virtual]`

Initialize converter.

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

See Also:

sosicon::IConverter::init()

Implements **sosicon::IConverter** (*p.34*).

Definition at line 31 of file converter_sosi2tsv.cpp.

void sosicon::ConverterSosi2tsv::run ()`[virtual]`

Start conversion.

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

See Also:

sosicon::IConverter::run()

Implements **sosicon::IConverter** (*p.34*).

Definition at line 36 of file converter_sosi2tsv.cpp.

Member Data Documentation

CommandLine sosicon::ConverterSosi2tsv::mCmd`[private]`

Definition at line 41 of file converter_sosi2tsv.h.

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

- /prosjekter/sosicon/src/**converter_sosi2tsv.h**
- /prosjekter/sosicon/src/**converter_sosi2tsv.cpp**

sosicon::CoordSys Class Reference

Coordinate system.

```
#include <coord_sys.h>
```

Public Member Functions

- **CoordSys** (std::string id)
- virtual **~CoordSys** ()
- std::string **getSosiId** ()
  Retrieve mSosiId.
- std::string **getDescription** ()
  Retrieve mDescription.
- std::string **getDatum** ()
  Retrieve mDatum.
- std::string **getProjection** ()
  Retrieve mProjection.
- std::string **getZone** ()
  Retrieve mZone.
- std::string **getSrid** ()
  Retrieve mSrid.

Private Attributes

- int **mId**
  KOORDSYS id.
- std::string **mSosiId**
  KOORDSYS id.
- std::string **mSrid**
  SRID.
- std::string **mDescription**
  Description of current reference.
- std::string **mDatum**
  Grid datum.
- std::string **mProjection**
  Map projection.
- std::string **mZone**
  UTM zone / axis.

Detailed Description

Coordinate system.

Resolves SOSI KOORDSYS codes and converts them to corresponting SRID identifiers if applicable. In SOSI, KOORDSYS is the equivalent to SRID, describing a set of predefined combinations of unit, datum and projection etc.

Definition at line 36 of file coord_sys.h.

Constructor & Destructor Documentation

sosicon::CoordSys::CoordSys (std::string *id*)

Definition at line 21 of file coord_sys.cpp.

sosicon::CoordSys::~CoordSys () `[virtual]`

Definition at line 351 of file coord_sys.cpp.

Member Function Documentation

std::string sosicon::CoordSys::getDatum () `[inline]`

Retrieve mDatum.

See Also:

sosicon::CoordSys::mDatum

Definition at line 108 of file coord_sys.h.

std::string sosicon::CoordSys::getDescription () `[inline]`

Retrieve mDescription.

See Also:

sosicon::CoordSys::mDescription

Definition at line 102 of file coord_sys.h.

std::string sosicon::CoordSys::getProjection () `[inline]`

Retrieve mProjection.

See Also:

sosicon::CoordSys::mProjection

Definition at line 114 of file coord_sys.h.

std::string sosicon::CoordSys::getSosiId () `[inline]`

Retrieve mSosiId.

See Also:

sosicon::CoordSys::mSosiId

Definition at line 96 of file coord_sys.h.

std::string sosicon::CoordSys::getSrid () `[inline]`

Retrieve mSrid.

See Also:

sosicon::CoordSys::mSrid

Definition at line 126 of file coord_sys.h.

std::string sosicon::CoordSys::getZone () `[inline]`

Retrieve mZone.

See Also:

sosicon::CoordSys::mZone

Definition at line 120 of file coord_sys.h.

Member Data Documentation

std::string sosicon::CoordSys::mDatum `[private]`

Grid datum.

Datum of current grid reference system. For instance "WGS84".

Definition at line 73 of file coord_sys.h.

std::string sosicon::CoordSys::mDescription `[private]`

Description of current reference.

Textual description of current KOORDSYS (grid reference system), as stated in the SOSI standard.

Definition at line 67 of file coord_sys.h.

int sosicon::CoordSys::mId `[private]`

KOORDSYS id.

The SOSI identifier for current reference system, named KOORDSYS in the file head.

Definition at line 44 of file coord_sys.h.

std::string sosicon::CoordSys::mProjection `[private]`

Map projection.

Description of the current projection. For instance "Gauss-Krüger".

Definition at line 79 of file coord_sys.h.

std::string sosicon::CoordSys::mSosiId `[private]`

KOORDSYS id.

String representation of the SOSI identifier for current reference system, named KOORDSYS in the file head.

Definition at line 51 of file coord_sys.h.

std::string sosicon::CoordSys::mSrid `[private]`

SRID.

Spatial reference system identifier, translation of the SOSI KOORDSYS ID. If no SRID exists for current KOORDSYS, this field will be empty. The SRID is a widely adopted standard for referencing various combinations of coordinate system, projection and datum.

Definition at line 60 of file coord_sys.h.

std::string sosicon::CoordSys::mZone `[private]`

UTM zone / axis.

The NGO axis or the UTM zone, dependent of current reference system.

Definition at line 85 of file coord_sys.h.

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

- /prosjekter/sosicon/src/**coord_sys.h**
- /prosjekter/sosicon/src/**coord_sys.cpp**

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 35 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:

converter	Reference to the pointer to receive the new **IConverter**.
cmd	**CommandLine** 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:

converter	Reference to a pointer containing the **IConverter** instance to be released. The pointer will be reset to 0 after object deletion.

Definition at line 33 of file factory.cpp.

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

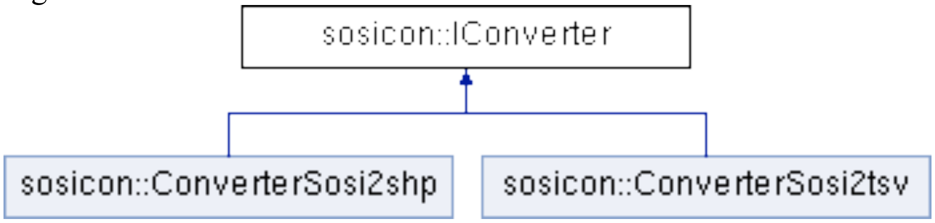
- /prosjekter/sosicon/src/**factory.h**
- /prosjekter/sosicon/src/**factory.cpp**

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** ()=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:

cmd	Arguments from the command-line parsed and ready within a **sosicon::CommandLine** object.

Implemented in **sosicon::ConverterSosi2shp** (*p.24*), and **sosicon::ConverterSosi2tsv** (*p.26*).

virtual void sosicon::IConverter::run ()`[pure virtual]`

Start conversion.

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

Implemented in **sosicon::ConverterSosi2shp** (*p.24*), and **sosicon::ConverterSosi2tsv** (*p.26*).

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

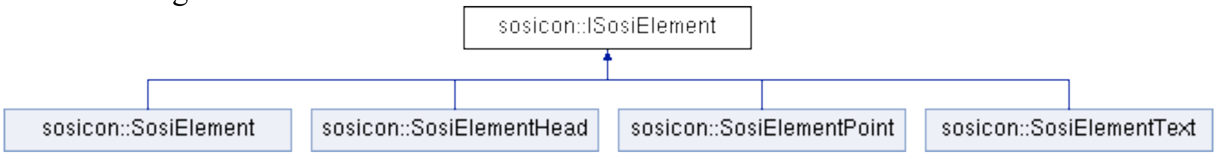
- /prosjekter/sosicon/src/**i_converter.h**

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::string **getData** (const char *key)=0
  Retrieve string data.
- virtual std::string **getData** (**CadastralUnit** *&cunit)=0
  Retrieve next cadastral unit.
- virtual std::string **getData** (**AddressUnit** *&aunit)=0
  Retrieve next address.
- virtual std::vector
- < std::string > & **getFields** ()=0
  Retrieve list of available SOSI fields.
- virtual std::string **getType** ()=0
  Retrieve SOSI element OBJTYPE.
- virtual void **set** (const std::string &key, const std::string &val)=0
  Set value.
- virtual void **append** (const std::string &key, char val)=0
  Append character to value.

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 retrieveing 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 49 of file i_sosi_element.h.

Constructor & Destructor Documentation

virtual sosicon::ISosiElement::~ISosiElement ()`[inline]`, `[virtual]`

Destructor.

Definition at line 53 of file i_sosi_element.h.

Member Function Documentation

virtual void sosicon::ISosiElement::append (const std::string & *key*, char *val*)`[pure virtual]`

Append character to value.

Appends one character to a SOSI field. The parser works by consuming characters one-by-one. Upon encountering a SOSI field value, the character is injected into the element's corresponding field through this function, placing the character at the end of the current string. If the field entry does not exist, it will be created.

See Also:

sosicon::Parser::parseSosiLine()

Parameters:

key	Name of the SOSI field to update.
var	The character to be appended at the end of current SOSI field value.

Implemented in **sosicon::SosiElementPoint** (*p.55*), **sosicon::SosiElement** (*p.47*), **sosicon::SosiElementHead** (*p.51*), and **sosicon::SosiElementText** (*p.61*).

virtual std::string sosicon::ISosiElement::getData (const char * *key*)`[pure virtual]`

Retrieve string data.

Get string value by key. Every SOSI element has several data fields. These fields are loaded into current element by the parser.

Parameters:

key	The key for the string data to be retrieved.

Returns:

If the key exists, the corresponding data string will be returned. Otherwise, a blank string is returned.

Implemented in **sosicon::SosiElementPoint** (*p.56*), **sosicon::SosiElement** (*p.48*), **sosicon::SosiElementHead** (*p.52*), and **sosicon::SosiElementText** (*p.62*).

virtual std::string sosicon::ISosiElement::getData (CadastralUnit *& *cunit*)`[pure virtual]`

Retrieve next cadastral unit.

If there are a vector of **CadastralUnit** objects associated with current **ISosiElement**, this function will return the next such object in the list. A pointer to the object will be stored in the provided cunit pointer reference.

If the initial value of the cunit pointer is 0, the first **CadastralUnit** in the list will be returned. Subsequent calls will retrieve the next item in list, until reaching the end. The pointer will then be set back to 0.

See Also:

sosicon::CadastralUnit

Parameters:

cunit	Reference to a pointer to receive the address to the **CadastralUnit** object.

Returns:

A string representation of the **CadastralUnit** being returned, or an empty string if the list was empty or the iterator moved past the last element in the list.

Implemented in **sosicon::SosiElementPoint** (*p.56*), **sosicon::SosiElement** (*p.48*), **sosicon::SosiElementHead** (*p.52*), and **sosicon::SosiElementText** (*p.62*).

virtual std::string sosicon::ISosiElement::getData (AddressUnit *& *aunit*)`[pure virtual]`

Retrieve next address.

If there are a vector of **AddressUnit** objects associated with current **ISosiElement**, this function will return the next such object in the list. A pointer to the object will be stored in the provided cunit pointer reference.

If the initial value of the aunit pointer is 0, the first **AddressUnit** in the list will be returned. Subsequent calls will retrieve the next item in list, until reaching the end. The pointer will then be set back to 0.

See Also:

sosicon::CadastralUnit

Parameters:

aunit	Reference to a pointer to receive the address to the **AddressUnit** object.

Returns:

A string representation of the **AddressUnit** being returned, or an empty string if the list was empty or the iterator moved past the last element in the list.

Implemented in **sosicon::SosiElementPoint** (*p.56*), **sosicon::SosiElement** (*p.47*), **sosicon::SosiElementHead** (*p.51*), and **sosicon::SosiElementText** (*p.61*).

virtual std::vector<std::string>& sosicon::ISosiElement::getFields ()`[pure virtual]`

Retrieve list of available SOSI fields.

Lists all SOSI fields from current element. These are the fields available for export to other formats.

Returns:

Reference to a string vector containing the names of all SOSI fields encountered by the parser in current element.

Implemented in **sosicon::SosiElementPoint** (*p.57*), **sosicon::SosiElement** (*p.48*), **sosicon::SosiElementHead** (*p.52*), and **sosicon::SosiElementText** (*p.62*).

virtual std::string sosicon::ISosiElement::getType () [pure virtual]

Retrieve SOSI element OBJTYPE.

Get the object type for current element.

Implemented in **sosicon::SosiElementPoint** (*p.57*), **sosicon::SosiElement** (*p.48*), **sosicon::SosiElementHead** (*p.52*), and **sosicon::SosiElementText** (*p.62*).

virtual void sosicon::ISosiElement::set (const std::string & *key*, const std::string & *val*) [pure virtual]

Set value.

Insert or update a SOSI field value. Call this function to insert string values into current element, or to alter an existing value. If the field entry does not exist, it will be created.

Parameters:

key	Name of the SOSI field to update/insert.
val	New string value.

Implemented in **sosicon::SosiElementPoint** (*p.57*), **sosicon::SosiElement** (*p.49*), **sosicon::SosiElementHead** (*p.53*), and **sosicon::SosiElementText** (*p.63*).

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

- /prosjekter/sosicon/src/**i_sosi_element.h**

sosicon::Parser Class Reference

SOSI file parser.

```
#include <parser.h>
```

Public Member Functions

- **Parser** ()
  Constructor.
- **~Parser** ()
  Destructor.
- bool **getNextSosiElement** (**ISosiElement** *&sosiElement)
  Retrieve next SOSI element for output.
- **ISosiElement * getHeadElement** ()
  Retrieve SOSI head.
- std::vector< std::string > & **getSosiElementsFields** ()
  Retrieve list of encountered SOSI fields.
- std::vector< std::string >
- ::size_type **getCountSosiElements** ()
  Get total number of SOSI elements.
- std::vector< std::string >
- ::size_type **getCountSosiElementsSelection** ()
  Get number of selected SOSI elements.
- void **setObjTypeSelection** (std::vector< std::string > &objTypes)
  Apply output selection filter.
- void **parseSosiLine** (std::string sosiLine)
  Main parser routine.
- void **dump** (bool all=false)
  Dump file content to stdout.
- void **reset** ()
  Reset parser, free allocated resources.

Private Attributes

- std::vector< **ISosiElement** * > **mSosiElements**
  SOSI elements container.
- std::vector< **ISosiElement** * > **mSosiElementsSelection**
  SOSI elements selected for output.
- std::vector< **ISosiElement** * >
- ::iterator **mSosiElementsIterator**
  *Current position within **Parser::mSosiElements**.*
- std::vector< std::string > **mSosiElementsFields**
  List containing all encountered SOSI fields in file.
- std::vector< std::string > **mObjTypeFilter**
  Object type filter.
- **ISosiElement * mHeadElement**
  Pointer to the head element.

- **ISosiElement * mCurrentElement**
  Pointer to last encountered SOSI element.

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_sosi_line.cpp** is generated on the basis of **parser/parser_sosi_line.rl** during pre-build processing.

Note:

Since **parser_sosi_line.cpp** is automatically re-generated before each compile, no redacting mat take place here. Any changes will be lost upon compile. Instead, its source script **parser/parser_sosi_line.rl** have to be edited.

Definition at line 51 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::dump (bool *all* = `false`)

Dump file content to stdout.

Writes a string representation of the SOSI file to stdout, for debugging purposes.

Parameters:

all	Dump all items in file. If this parameter is true, each SOSI element in the source file will be output. This may cause a lot of text to be output if the SOSI file is large.

Definition at line 47 of file parser.cpp.

std::vector< std::string >::size_type sosicon::Parser::getCountSosiElements ()

Get total number of SOSI elements.

Returns total count of SOSI elements minus the head element.

Returns:

The count of all SOSI elements except the head element.

Definition at line 130 of file parser.cpp.

std::vector< std::string >::size_type sosicon::Parser::getCountSosiElementsSelection ()

Get number of selected SOSI elements.

Returns number of SOSI elements scheduled for output. If no object type selection was specified, or the **Parser::mObjTypeFilter** list is emtpy, the count of all SOSI elements minus the head element will be returned.

See Also:

Parser::setObjTypeSelection(), **Parser::mObjTypeFilter**

Returns:

The number of SOSI elements scheduled for output.

Definition at line 135 of file parser.cpp.

sosicon::ISosiElement * sosicon::Parser::getHeadElement ()

Retrieve SOSI head.

The SOSI head element is special, since it contains global information about all other elements in the file.

This function returns a pointer to the head element.

See Also:

Parser::mHeadElement

Returns:

Pointer to a **ISosiElement** representing the file head.

Definition at line 161 of file parser.cpp.

bool sosicon::Parser::getNextSosiElement (ISosiElement *& *sosiElement*)

Retrieve next SOSI element for output.

Iterates through the list of SOSI elements to be exported for output.

Takes a reference to a **ISosiElement** pointer and assigns it to the next SOSI element in list. If the pointer is 0, the iteration will start at the beginning, otherwise it will continue to the next element, until reaching the end of the list.

If **Parser::mSosiElementsSelection** is populated, it will be promoted as the source list, thus emitting object types selected from the command-line only.

If **Parser::mSosiElementsSelection** is empty, **Parser::mSosiElements** will be used instad, emitting all encountered SOSI elements (except for the head element).

See Also:

Parser::mSosiElements, **Parser::mSosiElementsSelection**
Parser::mSosiElementsIterator

Note:

The pointer reference sosiElement must be initialized to zero before the first call to this function, but not between calls. It will be reset to zero once the iteration goes past the last element in list.

Example of usage:

```
ISosiElement* e = 0; // Must be initialized to zero!

// p is instance of Parser:
while( p.getNextSosiElement( e ) ) {

    std::cout << e->getType() << "\n";

}
```

Parameters:

sosiElement	Reference to a pointer receiving the address of the next element in list. If the pointer's initial value is zero, the iteration will start at the beginning of the list. When the iteration passes the last element, the pointer will be reset to zero.

Returns:

Status of the iteration. Use the return value in a while-loop to retrieve all values in sequence.

Return values:

true	Iteration is in progress. There are still more items to retrieve and the loop may go on.
false	The iteration reached the end of the list. There are no more items to retrieve and the loop must be terminated.

Definition at line 140 of file parser.cpp.

std::vector< std::string > & sosicon::Parser::getSosiElementsFields ()

Retrieve list of encountered SOSI fields.

Returns a summary of all fields encountered within current file. This function will iterate through every field in every SOSI element to build the field list. The list is lazy initialized and the build process is performed only once since it can be time consumng. Complexity is T(n) = O(n^2) and depends on the size of the input file.

Returns:

A reference to a string vector containing encountered field keys.

Definition at line 109 of file parser.cpp.

void sosicon::Parser::parseSosiLine (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 **parser/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:

sosiLine	Current line from the SOSI input file.

Definition at line 537 of file parser_sosi_line.cpp.

void sosicon::Parser::reset ()

Reset parser, free allocated resources.

Empties element lists, frees allocated resources, resets iterators and re-initializes internal pointers. After calling this function, the parser is ready to start consuming a new SOSI file from the beginning.

Definition at line 33 of file parser.cpp.

void sosicon::Parser::setObjTypeSelection (std::vector< std::string > & *objTypes*)

Apply output selection filter.

Which SOSI elements to export is governed by the **Parser::mObjTypeFilter** member. This string vector is initialized here. If the list contains one or more strings, only SOSI elements with a corresponding OBJTYPE will be scheduled for output.

The list of OBJTYPE strings are populated from the command-line interface, more specifically the -t parameter. It takes a comma-separated list of OBJTYPE names to specify which SOSI elements should be included in the target file. The comma-separated string is parsed and converted to a string vector by the **sosicon::CommandLine** class before it is passed on to the parser through this function.

See Also:

Parser::mObjTypeFilter, **sosicon::CommandLine::parse()**

Parameters:

objTypes	String vector containing the SOSI OBJTYPE identitifers for elements to include in the output.

Definition at line 125 of file parser.cpp.

Member Data Documentation

ISosiElement* sosicon::Parser::mCurrentElement`[private]`

Pointer to last encountered SOSI element.

Used internally by **Parser::parseSosiLine()** during file consumption.

The Ragel implemented parser consumes the SOSI file one line at the time. When encountering a new SOSI object, a corresponding **ISosiElement** implementation is created and placed in **Parser::mSosiElements** and possibly **Parser::mSosiElementsSelection**.

A pointer to the new object is assigned to this member in order to keep track of to the SOSI element currently in process.

See Also:

Parser::parseSosiLine()

Definition at line 128 of file parser.h.

ISosiElement* sosicon::Parser::mHeadElement `[private]`

Pointer to the head element.

The head element is not stored in the **Parser::mSosiElements** list because it contains data which is globally relevant to all elements in the SOSI file. Instead, it is available directly through this member. **Parser::getHeadElement()** provides external access to this member.

See Also:

Parser::getHeadElement()

Definition at line 113 of file parser.h.

std::vector<std::string> sosicon::Parser::mObjTypeFilter `[private]`

Object type filter.

SOSI object types scheduled for output. The vector is populated by the comma-separated values given by the command-line parameter -t. If the list is empty, all available objects will be exported without filtering.

See Also:

Parser::setObjTypeSelection()

Definition at line 102 of file parser.h.

std::vector<ISosiElement*> sosicon::Parser::mSosiElements `[private]`

SOSI elements container.

Stores pointers to all recognized and parsed SOSI file elements, except for the head element which is stored in **Parser::mHeadElement**. SOSI elements selected for output are pointed to by the items in **Parser::mSosiElementsSelection**.

Definition at line 59 of file parser.h.

std::vector<std::string> sosicon::Parser::mSosiElementsFields `[private]`

List containing all encountered SOSI fields in file.

Overview of all fields present in current SOSI file, regardless of object type.

When exporting to certain file formats, such as tab separated values (tsv), this field overview determines what columns should be present in the output file - unless a custom field list is specified via the command-line interface (sosicon command-line parameter -f).

See Also:

sosicon::ConverterSosi2tsv::run()

Definition at line 92 of file parser.h.

std::vector<ISosiElement*>::iterator sosicon::Parser::mSosiElementsIterator [private]

Current position within **Parser::mSosiElements**.

Used internally by **Parser::getNextSosiElement()**. Keeps track of current position while iterating the list of SOSI elements for output.

See Also:

Parser::getNextSosiElement()

Definition at line 79 of file parser.h.

std::vector<ISosiElement*> sosicon::Parser::mSosiElementsSelection [private]

SOSI elements selected for output.

Sosicon may schedule only certain SOSI object types for output, ignoring other object types in current file. This member variable keeps track of the elements matching the current object type filter. The filtering is determined by the contents of Parser::mObjTypes.

See Also:

mObjTypes

Definition at line 70 of file parser.h.

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

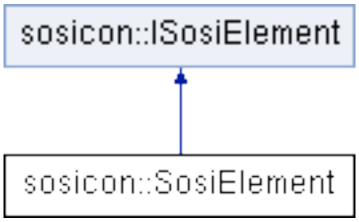
- /prosjekter/sosicon/src/**parser.h**
- /prosjekter/sosicon/src/**parser.cpp**
- /prosjekter/sosicon/src/**parser_sosi_line.cpp**
- /prosjekter/sosicon/src/parser/**parser_sosi_line.rl**

sosicon::SosiElement Class Reference

Basic SOSI element.

```
#include <sosi_element.h>
```

Inheritance diagram for sosicon::SosiElement:



Public Member Functions

- **SosiElement** ()
  Constructor.
- virtual **~SosiElement** ()
  Destructor.
- virtual std::string **getData** (**AddressUnit** *&aunit)
  *Get next **AddressUnit**.*
- virtual std::string **getData** (**CadastralUnit** *&cunit)
  *Get next **CadastralUnit**.*
- virtual std::string **getData** (const char *key)
  Retrieve string data.
- virtual std::vector
- < std::string > & **getFields** ()
  Retrieve list of available SOSI fields.
- virtual std::string **getType** ()
  Retrieve SOSI element OBJTYPE.
- virtual void **set** (const std::string &key, const std::string &val)
  Set value.
- virtual void **append** (const std::string &key, char val)
  Append character to value.

Private Attributes

- std::map< std::string,
- std::string > **mData**
  Mapped field values.
- std::vector< std::string > **mFields**
  List of field names.

Detailed Description

Basic SOSI element.

Implements basic characteristics of a SOSI element. Mostly a key/value container with **ISosiElement** stubs. Other SOSI elements delegates basic functionality to this class, whilst taking care of more specialized tasks themselves.

Definition at line 43 of file sosi_element.h.

Constructor & Destructor Documentation

sosicon::SosiElement::SosiElement ()

Constructor.

Definition at line 21 of file sosi_element.cpp.

sosicon::SosiElement::~SosiElement () `[virtual]`

Destructor.

Definition at line 24 of file sosi_element.cpp.

Member Function Documentation

void sosicon::SosiElement::append (const std::string & *key*, char *val*) `[virtual]`

Append character to value.

Appends one character to a SOSI field. The parser works by consuming characters one-by-one. Upon encountering a SOSI field value, the character is injected into the element's corresponding field through this function, placing the character at the end of the current string. If the field entry does not exist, it will be created.

See Also:

sosicon::Parser::parseSosiLine()

Parameters:

key	Name of the SOSI field to update.
var	The character to be appended at the end of current SOSI field value.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 32 of file sosi_element.cpp.

virtual std::string sosicon::SosiElement::getData (AddressUnit *& *aunit*) `[inline]`, `[virtual]`

Get next **AddressUnit**.

Not implemented in this class.

See Also:

ISosiElement::getData(AddressUnit*&)

Parameters:

aunit	Reference to pointer to **AddressUnit** will be set to 0.

Returns:

Empty string.

Implements **sosicon::ISosiElement** (*p.37*).

Definition at line 75 of file sosi_element.h.

virtual std::string sosicon::SosiElement::getData (CadastralUnit *& *cunit*)`[inline]`, `[virtual]`

Get next **CadastralUnit**.

Not implemented in this class.

See Also:

ISosiElement::getData(CadastralUnit*&)

Parameters:

cunit	Reference to pointer to **CadastralUnit** will be set to 0.

Returns:

Empty string.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 85 of file sosi_element.h.

std::string sosicon::SosiElement::getData (const char * *key*)`[virtual]`

Retrieve string data.

Get string value by key. Every SOSI element has several data fields. These fields are loaded into current element by the parser.

Parameters:

key	The key for the string data to be retrieved.

Returns:

If the key exists, the corresponding data string will be returned. Otherwise, a blank string is returned.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 37 of file sosi_element.cpp.

std::vector< std::string > & sosicon::SosiElement::getFields ()`[virtual]`

Retrieve list of available SOSI fields.

Lists all SOSI fields from current element. These are the fields available for export to other formats.

Returns:

Reference to a string vector containing the names of all SOSI fields encountered by the parser in current element.

Implements **sosicon::ISosiElement** (*p.37*).

Definition at line 42 of file sosi_element.cpp.

std::string sosicon::SosiElement::getType ()`[virtual]`

Retrieve SOSI element OBJTYPE.

Get the object type for current element.

Implements **sosicon::ISosiElement** (*p.38*).

Definition at line 51 of file sosi_element.cpp.

void sosicon::SosiElement::set (const std::string & *key*, const std::string & *val*)`[virtual]`

Set value.

Insert or update a SOSI field value. Call this function to insert string values into current element, or to alter an existing value. If the field entry does not exist, it will be created.

Parameters:

key	Name of the SOSI field to update/insert.
val	New string value.

Implements **sosicon::ISosiElement** (*p.38*).

Definition at line 27 of file sosi_element.cpp.

Member Data Documentation

std::map<std::string, std::string> sosicon::SosiElement::mData`[private]`

Mapped field values.

String vector containing textual data for current SOSI element. Key/value pairs are inserted by the parser.

Definition at line 50 of file sosi_element.h.

std::vector<std::string> sosicon::SosiElement::mFields`[private]`

List of field names.

String vector containing the keys (names) of all SOSI fields associated with current element. This is a list of data fields that may be included in the exported format.

Definition at line 57 of file sosi_element.h.

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

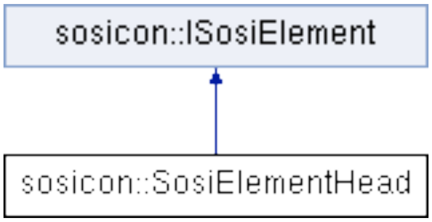
- /prosjekter/sosicon/src/**sosi_element.h**
- /prosjekter/sosicon/src/**sosi_element.cpp**

sosicon::SosiElementHead Class Reference

Sosi head element (HODE)

```
#include <sosi_element_head.h>
```

Inheritance diagram for sosicon::SosiElementHead:



Public Member Functions

- **SosiElementHead** ()
  Constructor.
- virtual **~SosiElementHead** ()
  Destructor.
- virtual std::string **getData** (**AddressUnit** *&aunit)
  *Get next **AddressUnit**.*
- virtual std::string **getData** (**CadastralUnit** *&cunit)
  *Get next **CadastralUnit**.*
- virtual std::string **getData** (const char *key)
  Retrieve string data.
- virtual std::vector
- < std::string > & **getFields** ()
  Retrieve list of available SOSI fields.
- virtual std::string **getType** ()
  Retrieve SOSI element OBJTYPE.
- virtual void **set** (const std::string &key, const std::string &val)
  Set value.
- virtual void **append** (const std::string &key, char val)
  Append character to value.

Private Attributes

- **SosiElement mElement**
  Base element.

Detailed Description

Sosi head element (HODE)

Represents the SOSI file header with global information about all other elements in file, such as metadata about coordinate system, scaling and transformation.

Definition at line 34 of file sosi_element_head.h.

Constructor & Destructor Documentation

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

Constructor.

Definition at line 44 of file sosi_element_head.h.

virtual sosicon::SosiElementHead::~SosiElementHead ()`[inline]`, `[virtual]`

Destructor.

Definition at line 47 of file sosi_element_head.h.

Member Function Documentation

virtual void sosicon::SosiElementHead::append (const std::string & *key*, char *val*)`[inline]`, `[virtual]`

Append character to value.

Appends one character to a SOSI field. The parser works by consuming characters one-by-one. Upon encountering a SOSI field value, the character is injected into the element's corresponding field through this function, placing the character at the end of the current string. If the field entry does not exist, it will be created.

See Also:

sosicon::Parser::parseSosiLine()

Parameters:

key	Name of the SOSI field to update.
var	The character to be appended at the end of current SOSI field value.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 82 of file sosi_element_head.h.

virtual std::string sosicon::SosiElementHead::getData (AddressUnit *& *aunit*)`[inline]`, `[virtual]`

Get next **AddressUnit**.

Not implemented in this class.

See Also:

ISosiElement::getData(AddressUnit*&)

Parameters:

aunit	Reference to pointer to **AddressUnit** will be set to 0.

Returns:

Empty string.

Implements **sosicon::ISosiElement** (*p.37*).

Definition at line 57 of file sosi_element_head.h.

virtual std::string sosicon::SosiElementHead::getData (CadastralUnit *& *cunit*)`[inline]`, `[virtual]`

Get next **CadastralUnit**.

Not implemented in this class.

See Also:

ISosiElement::getData(CadastralUnit*&)

Parameters:

cunit	Reference to pointer to **CadastralUnit** will be set to 0.

Returns:

Empty string.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 67 of file sosi_element_head.h.

virtual std::string sosicon::SosiElementHead::getData (const char * *key*)`[inline]`, `[virtual]`

Retrieve string data.

Get string value by key. Every SOSI element has several data fields. These fields are loaded into current element by the parser.

Parameters:

key	The key for the string data to be retrieved.

Returns:

If the key exists, the corresponding data string will be returned. Otherwise, a blank string is returned.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 70 of file sosi_element_head.h.

virtual std::vector<std::string>& sosicon::SosiElementHead::getFields ()`[inline]`, `[virtual]`

Retrieve list of available SOSI fields.

Lists all SOSI fields from current element. These are the fields available for export to other formats.

Returns:

Reference to a string vector containing the names of all SOSI fields encountered by the parser in current element.

Implements **sosicon::ISosiElement** (*p.37*).

Definition at line 73 of file sosi_element_head.h.

virtual std::string sosicon::SosiElementHead::getType ()`[inline]`, `[virtual]`

Retrieve SOSI element OBJTYPE.

Get the object type for current element.

Implements **sosicon::ISosiElement** (*p.38*).

Definition at line 76 of file sosi_element_head.h.

virtual void sosicon::SosiElementHead::set (const std::string & *key*, const std::string & *val*)`[inline]`, `[virtual]`

Set value.

Insert or update a SOSI field value. Call this function to insert string values into current element, or to alter an existing value. If the field entry does not exist, it will be created.

Parameters:

key	Name of the SOSI field to update/insert.
val	New string value.

Implements **sosicon::ISosiElement** (*p.38*).

Definition at line 79 of file sosi_element_head.h.

Member Data Documentation

SosiElement sosicon::SosiElementHead::mElement`[private]`

Base element.

Basic SOSI element functionality is delegated to this object.

Definition at line 40 of file sosi_element_head.h.

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

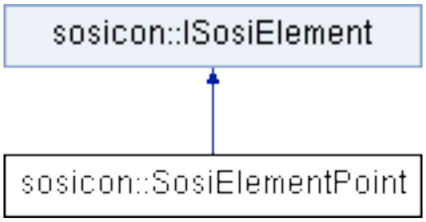
- /prosjekter/sosicon/src/**sosi_element_head.h**

sosicon::SosiElementPoint Class Reference

Sosi point element (PUNKT)

```
#include <sosi_element_point.h>
```

Inheritance diagram for sosicon::SosiElementPoint:



Public Member Functions

- **SosiElementPoint** ()
  Constructor.
- virtual **~SosiElementPoint** ()
  Destructor.
- virtual std::string **getData** (**AddressUnit** *&aunit)
  Retrieve next address.
- virtual std::string **getData** (**CadastralUnit** *&cunit)
  Retrieve next cadastral unit.
- virtual std::string **getData** (const char *key)
  Retrieve string data.
- virtual std::vector
- < std::string > & **getFields** ()
  Retrieve list of available SOSI fields.
- virtual std::string **getType** ()
  Retrieve SOSI element OBJTYPE.
- virtual void **set** (const std::string &key, const std::string &val)
  Set value.
- virtual void **append** (const std::string &key, char val)
  Append character to value.

Private Attributes

- **SosiElement mElement**
  Base element.
- std::vector< **CadastralUnit** * > **mCadastralUnits**
  List of cadastral units.
- std::vector< **CadastralUnit** * >
- ::iterator **mCadastralUnitIterator**
  Cadastral units iterator.
- **CadastralUnit** * **mCurrentCadastralUnit**
  *Pointer to current **CadastralUnit**.*
- std::vector< **AddressUnit** * > **mAddressUnits**
  List of address units.

- std::vector< **AddressUnit** * >
- ::iterator **mAddressUnitIterator**
  Address units iterator.
- **AddressUnit** * **mCurrentAddressUnit**
  *Pointer to current **AddressUnit**.*

Detailed Description

Sosi point element (PUNKT)

Represents the SOSI point object, describing a geographical location within current grid reference system.

Definition at line 36 of file sosi_element_point.h.

Constructor & Destructor Documentation

sosicon::SosiElementPoint::SosiElementPoint ()

Constructor.

Definition at line 21 of file sosi_element_point.cpp.

sosicon::SosiElementPoint::~SosiElementPoint () `[virtual]`

Destructor.

Definition at line 27 of file sosi_element_point.cpp.

Member Function Documentation

void sosicon::SosiElementPoint::append (const std::string & *key*, char *val*) `[virtual]`

Append character to value.

Appends one character to a SOSI field. The parser works by consuming characters one-by-one. Upon encountering a SOSI field value, the character is injected into the element's corresponding field through this function, placing the character at the end of the current string. If the field entry does not exist, it will be created.

See Also:

sosicon::Parser::parseSosiLine()

Parameters:

key	Name of the SOSI field to update.
var	The character to be appended at the end of current SOSI field value.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 41 of file sosi_element_point.cpp.

std::string sosicon::SosiElementPoint::getData (AddressUnit *& *aunit*)[virtual]

Retrieve next address.

If there are a vector of **AddressUnit** objects associated with current **ISosiElement**, this function will return the next such object in the list. A pointer to the object will be stored in the provided cunit pointer reference.

If the initial value of the aunit pointer is 0, the first **AddressUnit** in the list will be returned. Subsequent calls will retrieve the next item in list, until reaching the end. The pointer will then be set back to 0.

See Also:

sosicon::CadastralUnit

Parameters:

aunit	Reference to a pointer to receive the address to the **AddressUnit** object.

Returns:

A string representation of the **AddressUnit** being returned, or an empty string if the list was empty or the iterator moved past the last element in the list.

Implements **sosicon::ISosiElement** (*p.37*).

Definition at line 100 of file sosi_element_point.cpp.

std::string sosicon::SosiElementPoint::getData (CadastralUnit *& *cunit*)[virtual]

Retrieve next cadastral unit.

If there are a vector of **CadastralUnit** objects associated with current **ISosiElement**, this function will return the next such object in the list. A pointer to the object will be stored in the provided cunit pointer reference.

If the initial value of the cunit pointer is 0, the first **CadastralUnit** in the list will be returned. Subsequent calls will retrieve the next item in list, until reaching the end. The pointer will then be set back to 0.

See Also:

sosicon::CadastralUnit

Parameters:

cunit	Reference to a pointer to receive the address to the **CadastralUnit** object.

Returns:

A string representation of the **CadastralUnit** being returned, or an empty string if the list was empty or the iterator moved past the last element in the list.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 83 of file sosi_element_point.cpp.

virtual std::string sosicon::SosiElementPoint::getData (const char * *key*)[inline], [virtual]

Retrieve string data.

Get string value by key. Every SOSI element has several data fields. These fields are loaded into current element by the parser.

Parameters:

key	The key for the string data to be retrieved.

Returns:

If the key exists, the corresponding data string will be returned. Otherwise, a blank string is returned.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 103 of file sosi_element_point.h.

virtual std::vector<std::string>& sosicon::SosiElementPoint::getFields () `[inline]`, `[virtual]`

Retrieve list of available SOSI fields.

Lists all SOSI fields from current element. These are the fields available for export to other formats.

Returns:

Reference to a string vector containing the names of all SOSI fields encountered by the parser in current element.

Implements **sosicon::ISosiElement** (*p.37*).

Definition at line 106 of file sosi_element_point.h.

virtual std::string sosicon::SosiElementPoint::getType () `[inline]`, `[virtual]`

Retrieve SOSI element OBJTYPE.

Get the object type for current element.

Implements **sosicon::ISosiElement** (*p.38*).

Definition at line 109 of file sosi_element_point.h.

virtual void sosicon::SosiElementPoint::set (const std::string & *key*, const std::string & *val*) `[inline]`, `[virtual]`

Set value.

Insert or update a SOSI field value. Call this function to insert string values into current element, or to alter an existing value. If the field entry does not exist, it will be created.

Parameters:

key	Name of the SOSI field to update/insert.
val	New string value.

Implements **sosicon::ISosiElement** (*p.38*).

Definition at line 112 of file sosi_element_point.h.

Member Data Documentation

std::vector<AddressUnit*>::iterator sosicon::SosiElementPoint::mAddressUnitIterator `[private]`

Address units iterator.

Points at current location within **SosiElementPoint::mAddressUnits**. Used internally by **SosiElementPoint::getData(AddressUnit*&)**.

Definition at line 78 of file sosi_element_point.h.

std::vector<AddressUnit*> sosicon::SosiElementPoint::mAddressUnits `[private]`

List of address units.

If current SOSI element contains one or more related address units, they will be sequentially inserted into this vector.

Definition at line 71 of file sosi_element_point.h.

std::vector<CadastralUnit*>::iterator sosicon::SosiElementPoint::mCadastralUnitIterator `[private]`

Cadastral units iterator.

Points at current location within **SosiElementPoint::mCadastralUnits**. Used internally by **SosiElementPoint::getData(CadastralUnit*&)**.

Definition at line 56 of file sosi_element_point.h.

std::vector<CadastralUnit*> sosicon::SosiElementPoint::mCadastralUnits `[private]`

List of cadastral units.

If current SOSI element contains one or more related cadastral units, they will be sequentially inserted into this vector.

Definition at line 49 of file sosi_element_point.h.

AddressUnit* sosicon::SosiElementPoint::mCurrentAddressUnit `[private]`

Pointer to current **AddressUnit**.

Points to the **AddressUnit** currently in process by the parser. Used internally within **SosiElementPoint::append()** when the parser adds data to the last encountered **AddressUnit** element (i.e. the "ADRESSEBRUKSENHET" sub element within the SOSI file).

Definition at line 86 of file sosi_element_point.h.

CadastralUnit* sosicon::SosiElementPoint::mCurrentCadastralUnit `[private]`

Pointer to current **CadastralUnit**.

Points to the **CadastralUnit** currently in process by the parser. Used internally within **SosiElementPoint::append()** when the parser adds data to the last encountered **CadastralUnit** element (i.e. the "MATRIKKELNUMMER" sub element within the SOSI file).

Definition at line 64 of file sosi_element_point.h.

SosiElement sosicon::SosiElementPoint::mElement `[private]`

Base element.

Basic SOSI element functionality is delegated to this object.

Definition at line 42 of file sosi_element_point.h.

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

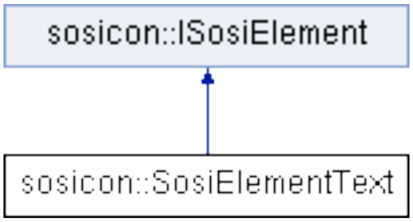
- /prosjekter/sosicon/src/**sosi_element_point.h**
- /prosjekter/sosicon/src/**sosi_element_point.cpp**

sosicon::SosiElementText Class Reference

Sosi text element (TEKST)

```
#include <sosi_element_text.h>
```

Inheritance diagram for sosicon::SosiElementText:



Public Member Functions

- **SosiElementText** ()
  Constructor.
- virtual **~SosiElementText** ()
  Destructor.
- virtual std::string **getData** (**AddressUnit** *&aunit)
  *Get next **AddressUnit**.*
- virtual std::string **getData** (**CadastralUnit** *&cunit)
  *Get next **CadastralUnit**.*
- virtual std::string **getData** (const char *key)
  Retrieve string data.
- virtual std::vector
- < std::string > & **getFields** ()
  Retrieve list of available SOSI fields.
- virtual std::string **getType** ()
  Retrieve SOSI element OBJTYPE.
- virtual void **set** (const std::string &key, const std::string &val)
  Set value.
- virtual void **append** (const std::string &key, char val)
  Append character to value.

Private Attributes

- **SosiElement mElement**
  Base element.

Detailed Description

Sosi text element (TEKST)

Represents the placement of a text string in the SOSI file specification.

Definition at line 33 of file sosi_element_text.h.

Constructor & Destructor Documentation

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

Constructor.

Definition at line 43 of file sosi_element_text.h.

virtual sosicon::SosiElementText::~SosiElementText ()`[inline]`, `[virtual]`

Destructor.

Definition at line 46 of file sosi_element_text.h.

Member Function Documentation

virtual void sosicon::SosiElementText::append (const std::string & *key*, char *val*)`[inline]`, `[virtual]`

Append character to value.

Appends one character to a SOSI field. The parser works by consuming characters one-by-one. Upon encountering a SOSI field value, the character is injected into the element's corresponding field through this function, placing the character at the end of the current string. If the field entry does not exist, it will be created.

See Also:

sosicon::Parser::parseSosiLine()

Parameters:

key	Name of the SOSI field to update.
var	The character to be appended at the end of current SOSI field value.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 81 of file sosi_element_text.h.

virtual std::string sosicon::SosiElementText::getData (AddressUnit *& *aunit*)`[inline]`, `[virtual]`

Get next **AddressUnit**.

Not implemented in this class.

See Also:

ISosiElement::getData(AddressUnit*&)

Parameters:

aunit	Reference to pointer to **AddressUnit** will be set to 0.

Returns:

Empty string.

Implements **sosicon::ISosiElement** (*p.37*).

Definition at line 56 of file sosi_element_text.h.

virtual std::string sosicon::SosiElementText::getData (CadastralUnit *& *cunit*) `[inline]`, `[virtual]`

Get next **CadastralUnit**.

Not implemented in this class.

See Also:

ISosiElement::getData(CadastralUnit*&)

Parameters:

cunit	Reference to pointer to **CadastralUnit** will be set to 0.

Returns:

Empty string.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 66 of file sosi_element_text.h.

virtual std::string sosicon::SosiElementText::getData (const char * *key*) `[inline]`, `[virtual]`

Retrieve string data.

Get string value by key. Every SOSI element has several data fields. These fields are loaded into current element by the parser.

Parameters:

key	The key for the string data to be retrieved.

Returns:

If the key exists, the corresponding data string will be returned. Otherwise, a blank string is returned.

Implements **sosicon::ISosiElement** (*p.36*).

Definition at line 69 of file sosi_element_text.h.

virtual std::vector<std::string>& sosicon::SosiElementText::getFields () `[inline]`, `[virtual]`

Retrieve list of available SOSI fields.

Lists all SOSI fields from current element. These are the fields available for export to other formats.

Returns:

Reference to a string vector containing the names of all SOSI fields encountered by the parser in current element.

Implements **sosicon::ISosiElement** (*p.37*).

Definition at line 72 of file sosi_element_text.h.

virtual std::string sosicon::SosiElementText::getType () `[inline]`, `[virtual]`

Retrieve SOSI element OBJTYPE.

Get the object type for current element.

Implements **sosicon::ISosiElement** (*p.38*).

Definition at line 75 of file sosi_element_text.h.

virtual void sosicon::SosiElementText::set (const std::string & *key*, const std::string & *val*)`[inline]`, `[virtual]`

Set value.

Insert or update a SOSI field value. Call this function to insert string values into current element, or to alter an existing value. If the field entry does not exist, it will be created.

Parameters:

key	Name of the SOSI field to update/insert.
val	New string value.

Implements **sosicon::ISosiElement** (*p.38*).

Definition at line 78 of file sosi_element_text.h.

Member Data Documentation

SosiElement sosicon::SosiElementText::mElement`[private]`

Base element.

Basic SOSI element functionality is delegated to this object.

Definition at line 39 of file sosi_element_text.h.

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

- /prosjekter/sosicon/src/**sosi_element_text.h**

File Documentation

/prosjekter/sosicon/src/address_unit.cpp File Reference

```
#include "address_unit.h"
```

/prosjekter/sosicon/src/address_unit.h File Reference

```
#include <string>
```

Classes

- class **sosicon::AddressUnit**

Address usage unit. Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/cadastral_unit.cpp File Reference

```
#include "cadastral_unit.h"
```

/prosjekter/sosicon/src/cadastral_unit.h File Reference

```
#include <string>
```

Classes

- class **sosicon::CadastralUnit**

Cadastral unit. Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/command_line.cpp File Reference

```
#include "command_line.h"
```

/prosjekter/sosicon/src/command_line.h File Reference

```
#include <iostream>
#include <vector>
#include <string>
#include "string_utils.h"
```

Classes

- class **sosicon::CommandLine**

Command-line parser. Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/converter_sosi2shp.cpp File Reference

```
#include "converter_sosi2shp.h"
```

/prosjekter/sosicon/src/converter_sosi2shp.h File Reference

```
#include <iostream>
#include <fstream>
#include <vector>
#include "i_converter.h"
#include "command_line.h"
#include "string_utils.h"
#include "parser.h"
```

Classes

- class **sosicon::ConverterSosi2shp**

SOSI to ESRI Shape converter. Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/converter_sosi2tsv.cpp File Reference

```
#include "converter_sosi2tsv.h"
```

/prosjekter/sosicon/src/converter_sosi2tsv.h File Reference

```
#include <iostream>
#include <fstream>
#include <vector>
#include "i_converter.h"
#include "command_line.h"
#include "parser.h"
```

Classes

- class **sosicon::ConverterSosi2tsv**

SOSI to TSV converter. Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/coord_sys.cpp File Reference

```
#include "coord_sys.h"
```

/prosjekter/sosicon/src/coord_sys.h File Reference

```
#include <string>
#include <sstream>
```

Classes

- class **sosicon::CoordSys**

Coordinate system. Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/factory.cpp File Reference

```
#include "factory.h"
```

/prosjekter/sosicon/src/factory.h File Reference

```
#include "i_converter.h"
#include "converter_sosi2shp.h"
#include "converter_sosi2tsv.h"
```

Classes

- class **sosicon::Factory**

Factory ***class.*** Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/i_converter.h File Reference

```
#include "command_line.h"
```

Classes

- class **sosicon::IConverter**

Interface: Converter. Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/i_sosi_element.h File Reference

```
#include <string>
#include <vector>
#include "address_unit.h"
#include "cadastral_unit.h"
```

Classes

- class **sosicon::ISosiElement**

Interface: SOSI element. Namespaces

- namespace **sosicon**

Application root.

/prosjekter/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.

/prosjekter/sosicon/src/main.h File Reference

```
#include <exception>
#include <iostream>
#include "command_line.h"
#include "factory.h"
#include "i_converter.h"
```

Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/parser.cpp File Reference

```
#include "parser.h"
```

/prosjekter/sosicon/src/parser.h File Reference

```
#include <vector>
#include <string>
#include <iostream>
#include "command_line.h"
#include "coord_sys.h"
#include "i_sosi_element.h"
#include "sosi_element_head.h"
#include "sosi_element_point.h"
#include "sosi_element_text.h"
```

Classes

- class **sosicon::Parser**

SOSI file parser. Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/parser/parser_sosi_line.rl File Reference

```
#include "parser.h"
```

Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/parser_pathinfo.cpp File Reference

```
#include "parser.h"
```

/prosjekter/sosicon/src/parser_sosi_line.cpp File Reference

```
#include "parser.h"
```

Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/sosi_element.cpp File Reference

```
#include "sosi_element.h"
```

/prosjekter/sosicon/src/sosi_element.h File Reference

```
#include <string>
#include <iostream>
#include <vector>
#include <map>
#include "i_sosi_element.h"
#include "coord_sys.h"
#include "address_unit.h"
#include "cadastral_unit.h"
```

Classes

- class **sosicon::SosiElement**

Basic SOSI element. Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/sosi_element_head.h File Reference

```
#include "sosi_element.h"
```

Classes

- class **sosicon::SosiElementHead**

Sosi head element (HODE) Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/sosi_element_point.cpp File Reference

```
#include "sosi_element_point.h"
```

/prosjekter/sosicon/src/sosi_element_point.h File Reference

```
#include "sosi_element.h"
#include "address_unit.h"
#include "cadastral_unit.h"
```

Classes

- class **sosicon::SosiElementPoint**

Sosi point element (PUNKT) Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/sosi_element_text.h File Reference

```
#include "sosi_element.h"
```

Classes

- class **sosicon::SosiElementText**

Sosi text element (TEKST) Namespaces

- namespace **sosicon**

Application root.

/prosjekter/sosicon/src/string_utils.cpp File Reference

```
#include "string_utils.h"
```

/prosjekter/sosicon/src/string_utils.h File Reference

```
#include "memory.h"
#include <string>
#include <ctype.h>
```

Namespaces

- namespace **sosicon**
- *Application root.* namespace **sosicon::stringUtils**

String manipulation routines. Functions

- std::string **sosicon::stringUtils::className2FileName** (const std::string &className)
  Converts Class name to file name string.
- std::string **sosicon::stringUtils::normalizeAppClassName** (const std::string &className)
  Asserts correct name of application classes.
- std::string **sosicon::stringUtils::repeat** (const std::string &seq, unsigned int count)
  Repeat string N times.
- std::string **sosicon::stringUtils::replaceAll** (const std::string &from, const std::string &to, const std::string &subject)
  Replace all occurences of one string witn another.
- std::string **sosicon::stringUtils::trim** (const std::string &str)
  Removes leading and trailing space characters.
- std::string **sosicon::stringUtils::trimLeft** (const std::string &str)
- std::string **sosicon::stringUtils::trimRight** (const std::string &str)
- std::string **sosicon::stringUtils::toLower** (const std::string &from)
- std::string **sosicon::stringUtils::ucFirst** (const std::string &str)
- void **sosicon::stringUtils::getPathInfo** (std::string path, std::string &dir, std::string &tit, std::string &ext)

Index

/prosjekter/sosicon/src/address_unit.cpp, 64
/prosjekter/sosicon/src/address_unit.h, 65
/prosjekter/sosicon/src/cadastral_unit.cpp, 66
/prosjekter/sosicon/src/cadastral_unit.h, 67
/prosjekter/sosicon/src/command_line.cpp, 68
/prosjekter/sosicon/src/command_line.h, 69
/prosjekter/sosicon/src/converter_sosi2shp.cpp, 70
/prosjekter/sosicon/src/converter_sosi2shp.h, 71
/prosjekter/sosicon/src/converter_sosi2tsv.cpp, 72
/prosjekter/sosicon/src/converter_sosi2tsv.h, 73
/prosjekter/sosicon/src/coord_sys.cpp, 74
/prosjekter/sosicon/src/coord_sys.h, 75
/prosjekter/sosicon/src/factory.cpp, 76
/prosjekter/sosicon/src/factory.h, 77
/prosjekter/sosicon/src/i_converter.h, 78
/prosjekter/sosicon/src/i_sosi_element.h, 79
/prosjekter/sosicon/src/main.cpp, 80
/prosjekter/sosicon/src/main.h, 81
/prosjekter/sosicon/src/parser_pathinfo.cpp, 85
/prosjekter/sosicon/src/parser_sosi_line.cpp, 86
/prosjekter/sosicon/src/parser.cpp, 82
/prosjekter/sosicon/src/parser.h, 83
/prosjekter/sosicon/src/parser/parser_sosi_line.rl, 84
/prosjekter/sosicon/src/sosi_element_head.h, 89
/prosjekter/sosicon/src/sosi_element_point.cpp, 90
/prosjekter/sosicon/src/sosi_element_point.h, 91
/prosjekter/sosicon/src/sosi_element_text.h, 92
/prosjekter/sosicon/src/sosi_element.cpp, 87
/prosjekter/sosicon/src/sosi_element.h, 88
/prosjekter/sosicon/src/string_utils.cpp, 93
/prosjekter/sosicon/src/string_utils.h, 94
~AddressUnit
  sosicon::AddressUnit, 14
~CadastralUnit
  sosicon::CadastralUnit, 17
~CommandLine
  sosicon::CommandLine, 20
~ConverterSosi2shp
  sosicon::ConverterSosi2shp, 23
~ConverterSosi2tsv
  sosicon::ConverterSosi2tsv, 25
~CoordSys
  sosicon::CoordSys, 28
~IConverter
  sosicon::IConverter, 33
~ISosiElement
  sosicon::ISosiElement, 36
~Parser
  sosicon::Parser, 40
~SosiElement
  sosicon::SosiElement, 47
~SosiElementHead
  sosicon::SosiElementHead, 51
~SosiElementPoint
  sosicon::SosiElementPoint, 55
~SosiElementText
  sosicon::SosiElementText, 61
AddressUnit
  sosicon::AddressUnit, 13
append
  sosicon::ISosiElement, 36
  sosicon::SosiElement, 47
  sosicon::SosiElementHead, 51
  sosicon::SosiElementPoint, 55
  sosicon::SosiElementText, 61
CadastralUnit
  sosicon::CadastralUnit, 17
className2FileName
  sosicon::stringUtils, 10
CommandLine
  sosicon::CommandLine, 20
Converters, 7
ConverterSosi2shp
  sosicon::ConverterSosi2shp, 23
ConverterSosi2tsv
  sosicon::ConverterSosi2tsv, 25
CoordSys
  sosicon::CoordSys, 28
dump
  sosicon::Parser, 40
get
  sosicon::Factory, 31
getCountSosiElements
  sosicon::Parser, 41
getCountSosiElementsSelection
  sosicon::Parser, 41
getData
  sosicon::ISosiElement, 36, 37
  sosicon::SosiElement, 47, 48
  sosicon::SosiElementHead, 51, 52
  sosicon::SosiElementPoint, 56
  sosicon::SosiElementText, 61, 62
getDatum
  sosicon::CoordSys, 28
getDescription
  sosicon::CoordSys, 28
getFields
  sosicon::ISosiElement, 37
  sosicon::SosiElement, 48
  sosicon::SosiElementHead, 52
  sosicon::SosiElementPoint, 57
  sosicon::SosiElementText, 62
getHeadElement
  sosicon::Parser, 41

getNextSosiElement
sosicon::Parser, 41
getPathInfo
sosicon::stringUtils, 10
getProjection
sosicon::CoordSys, 28
getSosiElementsFields
sosicon::Parser, 42
getSosiId
sosicon::CoordSys, 28
getSrid
sosicon::CoordSys, 29
getType
sosicon::ISosiElement, 38
sosicon::SosiElement, 48
sosicon::SosiElementHead, 52
sosicon::SosiElementPoint, 57
sosicon::SosiElementText, 62
getZone
sosicon::CoordSys, 29
GIS entities, 7
init
sosicon::ConverterSosi2shp, 24
sosicon::ConverterSosi2tsv, 26
sosicon::IConverter, 34
Interfaces, 7
mAddressUnitIterator
sosicon::SosiElementPoint, 57
mAddressUnits
sosicon::SosiElementPoint, 58
main
main.cpp, 80
main.cpp
main, 80
mAppend
sosicon::CommandLine, 20
mCadastralUnitIterator
sosicon::SosiElementPoint, 58
mCadastralUnits
sosicon::SosiElementPoint, 58
mCadastre
sosicon::CadastralUnit, 17
mCmd
sosicon::ConverterSosi2shp, 24
sosicon::ConverterSosi2tsv, 26
mCommand
sosicon::CommandLine, 21
mCurrentAddressUnit
sosicon::SosiElementPoint, 58
mCurrentCadastralUnit
sosicon::SosiElementPoint, 58
mCurrentElement
sosicon::Parser, 43
mData
sosicon::SosiElement, 49
mDatum
sosicon::CoordSys, 29
mDescription
sosicon::CoordSys, 29
mDestinationDirectory
sosicon::CommandLine, 21
mDoorNumber
sosicon::AddressUnit, 14
mElement
sosicon::SosiElementHead, 53
sosicon::SosiElementPoint, 58
sosicon::SosiElementText, 63
mFields
sosicon::SosiElement, 49
mFieldSelection
sosicon::CommandLine, 21
mFloorLevel
sosicon::AddressUnit, 14
mFloorNumber
sosicon::AddressUnit, 14
mHeadElement
sosicon::Parser, 44
mId
sosicon::CoordSys, 29
mIncludeHeader
sosicon::CommandLine, 21
mLeasehold
sosicon::CadastralUnit, 17
mMunicipality
sosicon::CadastralUnit, 17
mObjTypeFilter
sosicon::Parser, 44
mObjTypes
sosicon::CommandLine, 21
mOutputFile
sosicon::CommandLine, 21
mProjection
sosicon::CoordSys, 29
mProperty
sosicon::CadastralUnit, 17
mSection
sosicon::CadastralUnit, 18
mSosiElements
sosicon::Parser, 44
mSosiElementsFields
sosicon::Parser, 44
mSosiElementsIterator
sosicon::Parser, 45
mSosiElementsSelection
sosicon::Parser, 45
mSosiId
sosicon::CoordSys, 30
mSourceFiles
sosicon::CommandLine, 22
mSrid
sosicon::CoordSys, 30
mVerbose

sosicon::CommandLine, 22
mZone
sosicon::CoordSys, 30
normalizeAppClassName
sosicon::stringUtils, 11
outputHelpText
sosicon::CommandLine, 20
parse
sosicon::CommandLine, 20
Parser
sosicon::Parser, 40
parseSosiLine
sosicon::Parser, 43
release
sosicon::Factory, 31
repeat
sosicon::stringUtils, 11
replaceAll
sosicon::stringUtils, 11
reset
sosicon::Parser, 43
run
sosicon::ConverterSosi2shp, 24
sosicon::ConverterSosi2tsv, 26
sosicon::IConverter, 34
set
sosicon::ISosiElement, 38
sosicon::SosiElement, 49
sosicon::SosiElementHead, 53
sosicon::SosiElementPoint, 57
sosicon::SosiElementText, 63
setObjTypeSelection
sosicon::Parser, 43
SOSI Elements, 8
sosicon, 9
sosicon::AddressUnit, 13
~AddressUnit, 14
AddressUnit, 13
mDoorNumber, 14
mFloorLevel, 14
mFloorNumber, 14
toString, 14
sosicon::CadastralUnit, 16
~CadastralUnit, 17
CadastralUnit, 17
mCadastre, 17
mLeasehold, 17
mMunicipality, 17
mProperty, 17
mSection, 18
toString, 17
sosicon::CommandLine, 19
~CommandLine, 20
CommandLine, 20
mAppend, 20
mCommand, 21
mDestinationDirectory, 21
mFieldSelection, 21
mIncludeHeader, 21
mObjTypes, 21
mOutputFile, 21
mSourceFiles, 22
mVerbose, 22
outputHelpText, 20
parse, 20
sosicon::ConverterSosi2shp, 23
~ConverterSosi2shp, 23
ConverterSosi2shp, 23
init, 24
mCmd, 24
run, 24
sosicon::ConverterSosi2tsv, 25
~ConverterSosi2tsv, 25
ConverterSosi2tsv, 25
init, 26
mCmd, 26
run, 26
sosicon::CoordSys, 27
~CoordSys, 28
CoordSys, 28
getDatum, 28
getDescription, 28
getProjection, 28
getSosiId, 28
getSrid, 29
getZone, 29
mDatum, 29
mDescription, 29
mId, 29
mProjection, 29
mSosiId, 30
mSrid, 30
mZone, 30
sosicon::Factory, 31
get, 31
release, 31
sosicon::IConverter, 33
~IConverter, 33
init, 34
run, 34
sosicon::ISosiElement, 35
~ISosiElement, 36
append, 36
getData, 36, 37
getFields, 37
getType, 38
set, 38
sosicon::Parser, 39
~Parser, 40
dump, 40
getCountSosiElements, 41
getCountSosiElementsSelection, 41

getHeadElement, 41
getNextSosiElement, 41
getSosiElementsFields, 42
mCurrentElement, 43
mHeadElement, 44
mObjTypeFilter, 44
mSosiElements, 44
mSosiElementsFields, 44
mSosiElementsIterator, 45
mSosiElementsSelection, 45
Parser, 40
parseSosiLine, 43
reset, 43
setObjTypeSelection, 43
sosicon::SosiElement, 46
~SosiElement, 47
append, 47
getData, 47, 48
getFields, 48
getType, 48
mData, 49
mFields, 49
set, 49
SosiElement, 47
sosicon::SosiElementHead, 50
~SosiElementHead, 51
append, 51
getData, 51, 52
getFields, 52
getType, 52
mElement, 53
set, 53
SosiElementHead, 51
sosicon::SosiElementPoint, 54
~SosiElementPoint, 55
append, 55
getData, 56
getFields, 57
getType, 57
mAddressUnitIterator, 57
mAddressUnits, 58
mCadastralUnitIterator, 58
mCadastralUnits, 58
mCurrentAddressUnit, 58
mCurrentCadastralUnit, 58
mElement, 58
set, 57
SosiElementPoint, 55
sosicon::SosiElementText, 60
~SosiElementText, 61
append, 61
getData, 61, 62
getFields, 62
getType, 62
mElement, 63
set, 63
SosiElementText, 61
sosicon::stringUtils, 10
className2FileName, 10
getPathInfo, 10
normalizeAppClassName, 11
repeat, 11
replaceAll, 11
toLower, 11
trim, 11
trimLeft, 12
trimRight, 12
ucFirst, 12
SosiElement
sosicon::SosiElement, 47
SosiElementHead
sosicon::SosiElementHead, 51
SosiElementPoint
sosicon::SosiElementPoint, 55
SosiElementText
sosicon::SosiElementText, 61
toLower
sosicon::stringUtils, 11
toString
sosicon::AddressUnit, 14
sosicon::CadastralUnit, 17
trim
sosicon::stringUtils, 11
trimLeft
sosicon::stringUtils, 12
trimRight
sosicon::stringUtils, 12
ucFirst
sosicon::stringUtils, 12