txtEngine

v3.0

Generated by Doxygen 1.7.4

Sat Aug 13 2011 14:54:08

Contents

Chapter 1

Deprecated List

Member TiXmlHandle::Element() const use ToElement. Return the handle as a TiXmlElement. This may return null.

Member TiXmlHandle::Node() const use ToNode. Return the handle as a TiXmlNode. This may return null.

Member TiXmlHandle::Text() const use ToText() Return the handle as a TiXmlText. This may return null.

Member TiXmlHandle::Unknown() const use ToUnknown() Return the handle as a TiXmlUnknown. This may return null.

Chapter 2

Class Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Class Documentation

4.1 Area Class Reference

Public Member Functions

- bool **has_description** (std::string desc_id)
- std::string **get_status** ()
- bool **has_current_desc** ()
- int **get_num_items** ()
- std::string **get_description** ()
- void **remove_item** (int index)
- void **add_item** (Item *new_item)
- Item * **get_item** (int index)
- std::string **get_id** ()
- bool **has_item** (std::string item_to_find)
- Item * **get_item** (std::string item_id)
- void **add_description** (StateDescriptor *desc)
- void **add_command** (AreaCommand *command_name)
- int **get_num_commands** ()
- AreaCommand * **get_command** (int index)
- AreaCommand * **has_command** (std::string command_name)
- int **get_num_descriptions** ()
- StateDescriptor * **get_descriptor** (int index)
- **Area** (const char *id, const char *desc_id, const char *status)

Protected Attributes

- std::vector< Item * > **items**
- int **num_items**
- int **num_descriptions**
- int **num_commands**

- std::string **status**
- std::string **id**
- std::string **curr_desc_id**
- std::vector< StateDescriptor * > **description**
- std::vector< AreaCommand * > **commands**

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

- /home/cshome/m/mabrams/345/txtEngine/Area.h

4.2 AreaCommand Class Reference

Public Member Functions

- **AreaCommand** (const char *callmeby, const char *areatomoveto, const char *status_command, const char *depends_command)
- std::string **get_depends** ()
- std::string **get_status** ()
- std::string **get_name** ()
- std::string **get_area** ()
- std::string **get_message** ()
- void **set_message** (const char *to_message)
- bool **find** (std::string to_find)

Protected Attributes

- std::string **name**
- std::string **status**
- std::string **message**
- std::string **depends**
- std::string **move_to_area**

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

- /home/cshome/m/mabrams/345/txtEngine/AreaCommand.h

4.3 Item Class Reference

Public Member Functions

- bool **has_description** (std::string desc_id)
- bool **has_current_desc** ()
- std::string **get_description** ()

- void **add_description** (StateDescriptor *desc)
- void **change_collectable** (bool flip)
- bool **is_collectable** ()
- std::string **get_id** ()
- int **get_num_commands** ()
- void **add_command** (ItemCommand *command_name)
- ItemCommand * **get_command** (int index)
- ItemCommand * **has_command** (std::string command_name)
- int **get_num_descriptions** ()
- StateDescriptor * **get_descriptor** (int index)
- void **state_change** (std::string to_change)
- **Item** (bool collect, const char *identifier, const char *initial_state)

Protected Attributes

- bool **collectable**
- int **num_descriptions**
- int **num_commands**
- std::string **id**
- std::string **curr_desc_id**
- std::vector< StateDescriptor * > **description**
- std::vector< ItemCommand * > **commands**

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

- /home/cshome/m/mabrams/345/txtEngine/Item.h

4.4 ItemCommand Class Reference

Public Member Functions

- **ItemCommand** (const char *callmeby, const char *state_mutator, bool chng_collec, bool collec_dep, const char *area_chng, const char *status_command, const char *depends)
- std::string **get_depends** ()
- bool **get_change_collect** ()
- bool **get_collect_dependent** ()
- std::string **get_area_change** ()
- std::string **get_status** ()
- std::string **get_message** ()
- std::string **get_name** ()
- std::string **get_state_change** ()
- void **set_message** (const char *to_message)

Protected Attributes

- std::string **name**
- std::string **state_change**
- std::string **message**
- std::string **area_change**
- std::string **depends**
- std::string **status**
- bool **change_collect**
- bool **collect_dependent**

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

- /home/cshome/m/mabrams/345/txtEngine/ItemCommand.h

4.5 StateDescriptor Class Reference

Public Member Functions

- **StateDescriptor** (const char ∗identifier)
- void **set_description** (const char ∗desc)
- std::string **get_id** ()
- std::string **get_description** ()

Protected Attributes

- std::string **id**
- std::string **description**

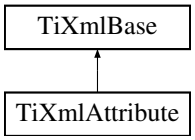
The documentation for this class was generated from the following file:

- /home/cshome/m/mabrams/345/txtEngine/StateDescriptor.h

4.6 TiXmlAttribute Class Reference

`#include <tinyxml.h>`

Inheritance diagram for TiXmlAttribute:



Public Member Functions

- TiXmlAttribute ()

  Construct an empty attribute.
- TiXmlAttribute (const char *_name, const char *_value)

  Construct an attribute with a name and value.
- const char * Name () const

  Return the name of this attribute.
- const char * Value () const

  Return the value of this attribute.
- int IntValue () const

  Return the value of this attribute, converted to an integer.
- double DoubleValue () const

  Return the value of this attribute, converted to a double.
- const TIXML_STRING & **NameTStr** () const
- int QueryIntValue (int *_value) const
- int QueryDoubleValue (double *_value) const

  QueryDoubleValue examines the value string. See QueryIntValue().
- void SetName (const char *_name)

  Set the name of this attribute.
- void SetValue (const char *_value)

  Set the value.
- void SetIntValue (int _value)

  Set the value from an integer.
- void SetDoubleValue (double _value)

  Set the value from a double.
- const TiXmlAttribute * Next () const

  Get the next sibling attribute in the DOM. Returns null at end.
- TiXmlAttribute * **Next** ()
- const TiXmlAttribute * Previous () const

  Get the previous sibling attribute in the DOM. Returns null at beginning.
- TiXmlAttribute * **Previous** ()
- bool **operator==** (const TiXmlAttribute &rhs) const
- bool **operator<** (const TiXmlAttribute &rhs) const
- bool **operator>** (const TiXmlAttribute &rhs) const
- virtual const char * **Parse** (const char *p, TiXmlParsingData *data, TiXmlEncoding encoding)
- virtual void Print (FILE *cfile, int depth) const
- void **Print** (FILE *cfile, int depth, TIXML_STRING *str) const
- void **SetDocument** (TiXmlDocument *doc)

Friends

- class TiXmlAttributeSet

4.6.1 Detailed Description

An attribute is a name-value pair. Elements have an arbitrary number of attributes, each with a unique name.

Note

The attributes are not TiXmlNodes, since they are not part of the tinyXML document object model. There are other suggested ways to look at this problem.

4.6.2 Member Function Documentation

4.6.2.1 virtual void TiXmlAttribute::Print (FILE * *cfile*, int *depth*) const [inline, virtual]

All TinyXml classes can print themselves to a filestream or the string class (TiXmlString in non-STL mode, std::string in STL mode.) Either or both cfile and str can be null.

This is a formatted print, and will insert tabs and newlines.

(For an unformatted stream, use the << operator.)

Implements TiXmlBase.

4.6.2.2 int TiXmlAttribute::QueryIntValue (int * *_value*) const

QueryIntValue examines the value string. It is an alternative to the IntValue() method with richer error checking. If the value is an integer, it is stored in 'value' and the call returns TIXML_SUCCESS. If it is not an integer, it returns TIXML_WRONG_TYPE.

A specialized but useful call. Note that for success it returns 0, which is the opposite of almost all other TinyXml calls.

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

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp
- /home/cshome/m/mabrams/345/txtEngine/tinyxmlparser.cpp

4.7 TiXmlAttributeSet Class Reference

Public Member Functions

- void **Add** (TiXmlAttribute *attribute)
- void **Remove** (TiXmlAttribute *attribute)
- const TiXmlAttribute * **First** () const
- TiXmlAttribute * **First** ()
- const TiXmlAttribute * **Last** () const
- TiXmlAttribute * **Last** ()

- TiXmlAttribute * **Find** (const char *_name) const
- TiXmlAttribute * **FindOrCreate** (const char *_name)

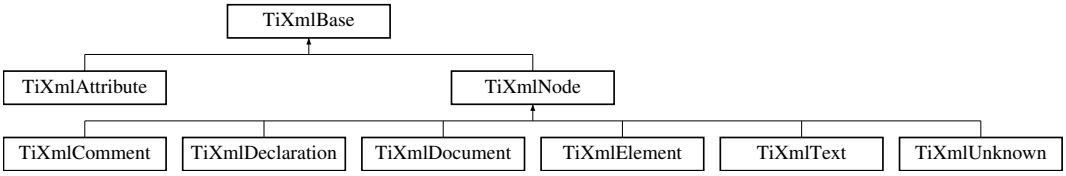
The documentation for this class was generated from the following files:

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp

4.8 TiXmlBase Class Reference

`#include <tinyxml.h>`

Inheritance diagram for TiXmlBase:



Classes

- struct **Entity**

Public Types

- enum {

  TIXML_NO_ERROR = 0, **TIXML_ERROR**, **TIXML_ERROR_OPENING_FILE**, **TIXML_-ERROR_PARSING_ELEMENT**,

  TIXML_ERROR_FAILED_TO_READ_ELEMENT_NAME, **TIXML_ERROR_READING_-ELEMENT_VALUE**, **TIXML_ERROR_READING_ATTRIBUTES**, **TIXML_ERROR_-PARSING_EMPTY**,

  TIXML_ERROR_READING_END_TAG, **TIXML_ERROR_PARSING_UNKNOWN**, **TIXML_ERROR_PARSING_COMMENT**, **TIXML_ERROR_PARSING_DECLARATION**,

  TIXML_ERROR_DOCUMENT_EMPTY, **TIXML_ERROR_EMBEDDED_NULL**, **TIXML_-ERROR_PARSING_CDATA**, **TIXML_ERROR_DOCUMENT_TOP_ONLY**,

  TIXML_ERROR_STRING_COUNT }

Public Member Functions

- virtual void Print (FILE *cfile, int depth) const =0
- int Row () const

- int Column () const

  See Row()
- void SetUserData (void *user)

  Set a pointer to arbitrary user data.
- void * GetUserData ()

  Get a pointer to arbitrary user data.
- const void * GetUserData () const

  Get a pointer to arbitrary user data.
- virtual const char * **Parse** (const char *p, TiXmlParsingData *data, TiXmlEncoding encoding)=0

Static Public Member Functions

- static void SetCondenseWhiteSpace (bool condense)
- static bool IsWhiteSpaceCondensed ()

  Return the current white space setting.
- static void EncodeString (const TIXML_STRING &str, TIXML_STRING *out)

Static Public Attributes

- static const int **utf8ByteTable** [256]

Static Protected Member Functions

- static const char * **SkipWhiteSpace** (const char *, TiXmlEncoding encoding)
- static bool **IsWhiteSpace** (char c)
- static bool **IsWhiteSpace** (int c)
- static const char * **ReadName** (const char *p, TIXML_STRING *name, TiXmlEncoding encoding)
- static const char * **ReadText** (const char *in, TIXML_STRING *text, bool ignoreWhiteSpace, const char *endTag, bool ignoreCase, TiXmlEncoding encoding)
- static const char * **GetEntity** (const char *in, char *value, int *length, TiXmlEncoding encoding)
- static const char * **GetChar** (const char *p, char *_value, int *length, TiXmlEncoding encoding)
- static bool **StringEqual** (const char *p, const char *endTag, bool ignoreCase, TiXmlEncoding encoding)
- static int **IsAlpha** (unsigned char anyByte, TiXmlEncoding encoding)
- static int **IsAlphaNum** (unsigned char anyByte, TiXmlEncoding encoding)
- static int **ToLower** (int v, TiXmlEncoding encoding)
- static void **ConvertUTF32ToUTF8** (unsigned long input, char *output, int *length)

Protected Attributes

- TiXmlCursor **location**
- void * userData

  Field containing a generic user pointer.

Static Protected Attributes

- static const char * **errorString** [TIXML_ERROR_STRING_COUNT]

Friends

- class TiXmlNode
- class TiXmlElement
- class TiXmlDocument

4.8.1 Detailed Description

TiXmlBase is a base class for every class in TinyXml. It does little except to establish that TinyXml classes can be printed and provide some utility functions.

In XML, the document and elements can contain other elements and other types of nodes.

```
A Document can contain: Element (container or leaf)
Comment (leaf)
Unknown (leaf)
Declaration( leaf )

An Element can contain: Element (container or leaf)
Text (leaf)
Attributes (not on tree)
Comment (leaf)
Unknown (leaf)

A Decleration contains: Attributes (not on tree)
```

4.8.2 Member Function Documentation

4.8.2.1 void TiXmlBase::EncodeString (const TIXML_STRING & *str*, TIXML_STRING * *out*) [static]

Expands entities in a string. Note this should not contian the tag's '<', '>', etc, or they will be transformed into entities!

4.8.2.2 virtual void TiXmlBase::Print (FILE * *cfile,* int *depth*) const [pure virtual]

All TinyXml classes can print themselves to a filestream or the string class (TiXmlString in non-STL mode, std::string in STL mode.) Either or both cfile and str can be null.

This is a formatted print, and will insert tabs and newlines.

(For an unformatted stream, use the << operator.)

Implemented in TiXmlAttribute, TiXmlElement, TiXmlComment, TiXmlText, TiXmlDeclaration, TiXmlUnknown, and TiXmlDocument.

4.8.2.3 int TiXmlBase::Row ( ) const [inline]

Return the position, in the original source file, of this node or attribute. The row and column are 1-based. (That is the first row and first column is 1,1). If the returns values are 0 or less, then the parser does not have a row and column value.

Generally, the row and column value will be set when the TiXmlDocument::Load(), TiXmlDocument::LoadFile(), or any TiXmlNode::Parse() is called. It will NOT be set when the DOM was created from operator>>.

The values reflect the initial load. Once the DOM is modified programmatically (by adding or changing nodes and attributes) the new values will NOT update to reflect changes in the document.

There is a minor performance cost to computing the row and column. Computation can be disabled if TiXmlDocument::SetTabSize() is called with 0 as the value.

See also

TiXmlDocument::SetTabSize()

4.8.2.4 static void TiXmlBase::SetCondenseWhiteSpace (bool *condense*) [inline, static]

The world does not agree on whether white space should be kept or not. In order to make everyone happy, these global, static functions are provided to set whether or not TinyXml will condense all white space into a single space or not. The default is to condense. Note changing this value is not thread safe.

4.8.3 Member Data Documentation

4.8.3.1 const char * TiXmlBase::errorString [static, protected]

Initial value:

```
{
        "No error",
        "Error",
        "Failed to open file",
```

```
        "Error parsing Element.",
        "Failed to read Element name",
        "Error reading Element value.",
        "Error reading Attributes.",
        "Error: empty tag.",
        "Error reading end tag.",
        "Error parsing Unknown.",
        "Error parsing Comment.",
        "Error parsing Declaration.",
        "Error document empty.",
        "Error null (0) or unexpected EOF found in input stream.",
        "Error parsing CDATA.",
        "Error when TiXmlDocument added to document, because TiXmlDocument can on
      ly be at the root.",
}
```

4.8.3.2 const int TiXmlBase::utf8ByteTable [static]

Initial value:

```
{
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      1,      1,      1,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1,
                1,      1,      2,      2,      2,      2,      2,      2,      2
      ,       2,      2,      2,      2,      2,      2,      2,
                2,      2,      2,      2,      2,      2,      2,      2,      2
      ,       2,      2,      2,      2,      2,      2,      2,
                3,      3,      3,      3,      3,      3,      3,      3,      3
      ,       3,      3,      3,      3,      3,      3,      3,
                4,      4,      4,      4,      4,      1,      1,      1,      1
      ,       1,      1,      1,      1,      1,      1,      1
}
```

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

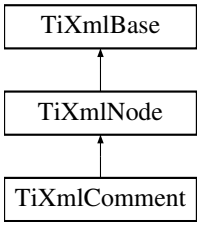
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp
- /home/cshome/m/mabrams/345/txtEngine/tinyxmlerror.cpp
- /home/cshome/m/mabrams/345/txtEngine/tinyxmlparser.cpp

4.9 TiXmlComment Class Reference

`#include <tinyxml.h>`

Inheritance diagram for TiXmlComment:



Public Member Functions

- TiXmlComment ()

  Constructs an empty comment.
- TiXmlComment (const char *_value)

  Construct a comment from text.
- **TiXmlComment** (const TiXmlComment &)
- TiXmlComment & **operator=** (const TiXmlComment &base)
- virtual TiXmlNode * Clone () const

  Returns a copy of this Comment.
- virtual void Print (FILE *cfile, int depth) const
- virtual const char * **Parse** (const char *p, TiXmlParsingData *data, TiXmlEncoding encoding)
- virtual const TiXmlComment * ToComment () const

  Cast to a more defined type. Will return null not of the requested type.
- virtual TiXmlComment * ToComment ()

  Cast to a more defined type. Will return null not of the requested type.
- virtual bool Accept (TiXmlVisitor *visitor) const

Protected Member Functions

- void **CopyTo** (TiXmlComment *target) const

4.9.1 Detailed Description

An XML comment.

4.9.2 Member Function Documentation

4.9.2.1 bool TiXmlComment::Accept (TiXmlVisitor * *visitor*) const [virtual]

Walk the XML tree visiting this node and all of its children.

Implements TiXmlNode.

4.9.2.2 void TiXmlComment::Print (FILE * *cfile*, int *depth*) const [virtual]

All TinyXml classes can print themselves to a filestream or the string class (TiXmlString in non-STL mode, std::string in STL mode.) Either or both cfile and str can be null.

This is a formatted print, and will insert tabs and newlines.

(For an unformatted stream, use the << operator.)

Implements TiXmlBase.

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

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp
- /home/cshome/m/mabrams/345/txtEngine/tinyxmlparser.cpp

4.10 TiXmlCursor Struct Reference

Public Member Functions

- void **Clear** ()

Public Attributes

- int **row**
- int **col**

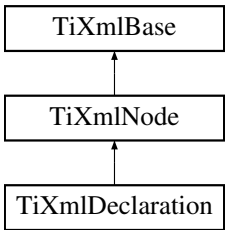
The documentation for this struct was generated from the following file:

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h

4.11 TiXmlDeclaration Class Reference

```
#include <tinyxml.h>
```

Inheritance diagram for TiXmlDeclaration:



Public Member Functions

- TiXmlDeclaration ()

  Construct an empty declaration.
- TiXmlDeclaration (const char *_version, const char *_encoding, const char *_standalone)

  Construct.
- **TiXmlDeclaration** (const TiXmlDeclaration &copy)
- TiXmlDeclaration & **operator=** (const TiXmlDeclaration &copy)
- const char * Version () const

  Version. Will return an empty string if none was found.
- const char * Encoding () const

  Encoding. Will return an empty string if none was found.
- const char * Standalone () const

  Is this a standalone document?
- virtual TiXmlNode * Clone () const

  Creates a copy of this Declaration and returns it.
- virtual void **Print** (FILE *cfile, int depth, TIXML_STRING *str) const
- virtual void Print (FILE *cfile, int depth) const
- virtual const char * **Parse** (const char *p, TiXmlParsingData *data, TiXmlEncoding encoding)
- virtual const TiXmlDeclaration * ToDeclaration () const

  Cast to a more defined type. Will return null not of the requested type.
- virtual TiXmlDeclaration * ToDeclaration ()

  Cast to a more defined type. Will return null not of the requested type.
- virtual bool Accept (TiXmlVisitor *visitor) const

Protected Member Functions

- void **CopyTo** (TiXmlDeclaration *target) const

4.11.1 Detailed Description

In correct XML the declaration is the first entry in the file.

```
<?xml version="1.0" standalone="yes"?>
```

TinyXml will happily read or write files without a declaration, however. There are 3 possible attributes to the declaration: version, encoding, and standalone.

Note: In this version of the code, the attributes are handled as special cases, not generic attributes, simply because there can only be at most 3 and they are always the same.

4.11.2 Member Function Documentation

4.11.2.1 bool TiXmlDeclaration::Accept (TiXmlVisitor * *visitor*) const [virtual]

Walk the XML tree visiting this node and all of its children.

Implements TiXmlNode.

4.11.2.2 virtual void TiXmlDeclaration::Print (FILE * *cfile,* int *depth*) const [inline, virtual]

All TinyXml classes can print themselves to a filestream or the string class (TiXmlString in non-STL mode, std::string in STL mode.) Either or both cfile and str can be null.

This is a formatted print, and will insert tabs and newlines.

(For an unformatted stream, use the << operator.)

Implements TiXmlBase.

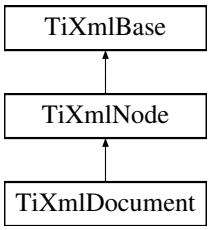
The documentation for this class was generated from the following files:

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp
- /home/cshome/m/mabrams/345/txtEngine/tinyxmlparser.cpp

4.12 TiXmlDocument Class Reference

```
#include <tinyxml.h>
```

Inheritance diagram for TiXmlDocument:



Public Member Functions

- TiXmlDocument ()

  Create an empty document, that has no name.
- TiXmlDocument (const char ∗documentName)

  Create a document with a name. The name of the document is also the filename of the xml.
- **TiXmlDocument** (const TiXmlDocument &copy)
- TiXmlDocument & **operator=** (const TiXmlDocument &copy)
- bool LoadFile (TiXmlEncoding encoding=TIXML_DEFAULT_ENCODING)
- bool SaveFile () const

  Save a file using the current document value. Returns true if successful.
- bool LoadFile (const char ∗filename, TiXmlEncoding encoding=TIXML_DEFAULT_ENCODING)

  Load a file using the given filename. Returns true if successful.
- bool SaveFile (const char ∗filename) const

  Save a file using the given filename. Returns true if successful.
- bool LoadFile (FILE ∗, TiXmlEncoding encoding=TIXML_DEFAULT_ENCODING)
- bool SaveFile (FILE ∗) const

  Save a file using the given FILE∗. Returns true if successful.
- virtual const char ∗ Parse (const char ∗p, TiXmlParsingData ∗data=0, TiXmlEncoding encoding=TIXML_DEFAULT_ENCODING)
- const TiXmlElement ∗ RootElement () const
- TiXmlElement ∗ **RootElement** ()
- bool Error () const
- const char ∗ ErrorDesc () const

  Contains a textual (english) description of the error if one occurs.
- int ErrorId () const
- int ErrorRow () const
- int ErrorCol () const

  The column where the error occured. See ErrorRow()
- void SetTabSize (int _tabsize)
- int **TabSize** () const
- void ClearError ()
- void Print () const
- virtual void Print (FILE ∗cfile, int depth=0) const

Print this Document to a FILE stream.

- void **SetError** (int err, const char *errorLocation, TiXmlParsingData *prevData, TiXmlEncoding encoding)
- virtual const TiXmlDocument * ToDocument () const

  Cast to a more defined type. Will return null not of the requested type.
- virtual TiXmlDocument * ToDocument ()

  Cast to a more defined type. Will return null not of the requested type.
- virtual bool Accept (TiXmlVisitor *content) const

Protected Member Functions

- virtual TiXmlNode * Clone () const

4.12.1 Detailed Description

Always the top level node. A document binds together all the XML pieces. It can be saved, loaded, and printed to the screen. The 'value' of a document node is the xml file name.

4.12.2 Member Function Documentation

4.12.2.1 bool TiXmlDocument::Accept (TiXmlVisitor * *content*) const [virtual]

Walk the XML tree visiting this node and all of its children.

Implements TiXmlNode.

4.12.2.2 void TiXmlDocument::ClearError ( ) [inline]

If you have handled the error, it can be reset with this call. The error state is automatically cleared if you Parse a new XML block.

4.12.2.3 TiXmlNode * TiXmlDocument::Clone ( ) const [protected, virtual]

Create an exact duplicate of this node and return it. The memory must be deleted by the caller.

Implements TiXmlNode.

4.12.2.4 bool TiXmlDocument::Error ( ) const [inline]

If an error occurs, Error will be set to true. Also,

- The ErrorId() will contain the integer identifier of the error (not generally useful)

- The ErrorDesc() method will return the name of the error. (very useful)
- The ErrorRow() and ErrorCol() will return the location of the error (if known)

4.12.2.5 int TiXmlDocument::ErrorId ( ) const [inline]

Generally, you probably want the error string (ErrorDesc()). But if you prefer the ErrorId, this function will fetch it.

4.12.2.6 int TiXmlDocument::ErrorRow ( ) const [inline]

Returns the location (if known) of the error. The first column is column 1, and the first row is row 1. A value of 0 means the row and column wasn't applicable (memory errors, for example, have no row/column) or the parser lost the error. (An error in the error reporting, in that case.)

See also

SetTabSize, Row, Column

4.12.2.7 bool TiXmlDocument::LoadFile (TiXmlEncoding *encoding* = TIXML_DEFAULT_ENCODING)

Load a file using the current document value. Returns true if successful. Will delete any existing document data before loading.

4.12.2.8 bool TiXmlDocument::LoadFile (FILE * *file*, TiXmlEncoding *encoding* = TIXML_DEFAULT_ENCODING)

Load a file using the given FILE*. Returns true if successful. Note that this method doesn't stream - the entire object pointed at by the FILE* will be interpreted as an XML file. TinyXML doesn't stream in XML from the current file location. Streaming may be added in the future.

4.12.2.9 const char * TiXmlDocument::Parse (const char * *p*, TiXmlParsingData * *data* = 0, TiXmlEncoding *encoding* = TIXML_DEFAULT_ENCODING) [virtual]

Parse the given null terminated block of xml data. Passing in an encoding to this method (either TIXML_ENCODING_LEGACY or TIXML_ENCODING_UTF8 will force TinyXml to use that encoding, regardless of what TinyXml might otherwise try to detect.

Implements TiXmlBase.

4.12.2.10 void TiXmlDocument::Print ( ) const [inline]

Write the document to standard out using formatted printing ("pretty print").

4.12.2.11 const TiXmlElement* TiXmlDocument::RootElement ( ) const [inline]

Get the root element -- the only top level element -- of the document. In well formed XML, there should only be one. TinyXml is tolerant of multiple elements at the document level.

4.12.2.12 void TiXmlDocument::SetTabSize (int *_tabsize*) [inline]

SetTabSize() allows the error reporting functions (ErrorRow() and ErrorCol()) to report the correct values for row and column. It does not change the output or input in any way.

By calling this method, with a tab size greater than 0, the row and column of each node and attribute is stored when the file is loaded. Very useful for tracking the DOM back in to the source file.

The tab size is required for calculating the location of nodes. If not set, the default of 4 is used. The tabsize is set per document. Setting the tabsize to 0 disables row/column tracking.

Note that row and column tracking is not supported when using operator>>.

The tab size needs to be enabled before the parse or load. Correct usage:

```
TiXmlDocument doc;
doc.SetTabSize( 8 );
doc.Load( "myfile.xml" );
```

See also

Row, Column

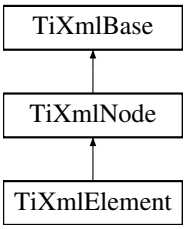
The documentation for this class was generated from the following files:

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp
- /home/cshome/m/mabrams/345/txtEngine/tinyxmlparser.cpp

4.13 TiXmlElement Class Reference

```
#include <tinyxml.h>
```

Inheritance diagram for TiXmlElement:



Public Member Functions

- TiXmlElement (const char *in_value)

  Construct an element.
- **TiXmlElement** (const TiXmlElement &)
- TiXmlElement & **operator=** (const TiXmlElement &base)
- const char * Attribute (const char *name) const
- const char * Attribute (const char *name, int *i) const
- const char * Attribute (const char *name, double *d) const
- int QueryIntAttribute (const char *name, int *_value) const
- int QueryUnsignedAttribute (const char *name, unsigned *_value) const

  QueryUnsignedAttribute examines the attribute - see QueryIntAttribute().
- int QueryBoolAttribute (const char *name, bool *_value) const
- int QueryDoubleAttribute (const char *name, double *_value) const

  QueryDoubleAttribute examines the attribute - see QueryIntAttribute().
- int QueryFloatAttribute (const char *name, float *_value) const

  QueryFloatAttribute examines the attribute - see QueryIntAttribute().
- void SetAttribute (const char *name, const char *_value)
- void SetAttribute (const char *name, int value)
- void SetDoubleAttribute (const char *name, double value)
- void RemoveAttribute (const char *name)
- const TiXmlAttribute * FirstAttribute () const

  Access the first attribute in this element.
- TiXmlAttribute * **FirstAttribute** ()
- const TiXmlAttribute * LastAttribute () const

  Access the last attribute in this element.
- TiXmlAttribute * **LastAttribute** ()
- const char * GetText () const
- virtual TiXmlNode * Clone () const

  Creates a new Element and returns it - the returned element is a copy.
- virtual void Print (FILE *cfile, int depth) const
- virtual const char * **Parse** (const char *p, TiXmlParsingData *data, TiXmlEncoding encoding)
- virtual const TiXmlElement * ToElement () const

  Cast to a more defined type. Will return null not of the requested type.
- virtual TiXmlElement * ToElement ()

  Cast to a more defined type. Will return null not of the requested type.
- virtual bool Accept (TiXmlVisitor *visitor) const

Protected Member Functions

- void **CopyTo** (TiXmlElement *target) const
- void **ClearThis** ()
- const char * **ReadValue** (const char *in, TiXmlParsingData *prevData, TiXmlEncoding encoding)

4.13.1 Detailed Description

The element is a container class. It has a value, the element name, and can contain other elements, text, comments, and unknowns. Elements also contain an arbitrary number of attributes.

4.13.2 Member Function Documentation

4.13.2.1 bool TiXmlElement::Accept (TiXmlVisitor * *visitor*) const [virtual]

Walk the XML tree visiting this node and all of its children.

Implements TiXmlNode.

4.13.2.2 const char * TiXmlElement::Attribute (const char * *name*, double * *d*) const

Given an attribute name, Attribute() returns the value for the attribute of that name, or null if none exists. If the attribute exists and can be converted to an double, the double value will be put in the return 'd', if 'd' is non-null.

4.13.2.3 const char * TiXmlElement::Attribute (const char * *name*) const

Given an attribute name, Attribute() returns the value for the attribute of that name, or null if none exists.

4.13.2.4 const char * TiXmlElement::Attribute (const char * *name*, int * *i*) const

Given an attribute name, Attribute() returns the value for the attribute of that name, or null if none exists. If the attribute exists and can be converted to an integer, the integer value will be put in the return 'i', if 'i' is non-null.

4.13.2.5 const char * TiXmlElement::GetText ( ) const

Convenience function for easy access to the text inside an element. Although easy and concise, GetText() is limited compared to getting the TiXmlText child and accessing it directly.

If the first child of 'this' is a TiXmlText, the GetText() returns the character string of the Text node, else null is returned.

This is a convenient method for getting the text of simple contained text:

```
<foo>This is text</foo>
const char* str = fooElement->GetText();
```

'str' will be a pointer to "This is text".

Note that this function can be misleading. If the element foo was created from this XML:

```
<foo><b>This is text</b></foo>
```

then the value of str would be null. The first child node isn't a text node, it is another element. From this XML:

```
<foo>This is <b>text</b></foo>
```

GetText() will return "This is ".

WARNING: GetText() accesses a child node - don't become confused with the similarly named TiXmlHandle::Text() and TiXmlNode::ToText() which are safe type casts on the referenced node.

4.13.2.6 void TiXmlElement::Print (FILE * *cfile*, int *depth*) const [virtual]

All TinyXml classes can print themselves to a filestream or the string class (TiXmlString in non-STL mode, std::string in STL mode.) Either or both cfile and str can be null.

This is a formatted print, and will insert tabs and newlines.

(For an unformatted stream, use the << operator.)

Implements TiXmlBase.

4.13.2.7 int TiXmlElement::QueryBoolAttribute (const char * *name*, bool * *_value*) const

QueryBoolAttribute examines the attribute - see QueryIntAttribute(). Note that '1', 'true', or 'yes' are considered true, while '0', 'false' and 'no' are considered false.

4.13.2.8 int TiXmlElement::QueryIntAttribute (const char * *name*, int * *_value*) const

QueryIntAttribute examines the attribute - it is an alternative to the Attribute() method with richer error checking. If the attribute is an integer, it is stored in 'value' and the call returns TIXML_SUCCESS. If it is not an integer, it returns TIXML_WRONG_TYPE. If the attribute does not exist, then TIXML_NO_ATTRIBUTE is returned.

4.13.2.9 void TiXmlElement::RemoveAttribute (const char * *name*)

Deletes an attribute with the given name.

4.13.2.10 void TiXmlElement::SetAttribute (const char * *name*, const char * *_value*)

Sets an attribute of name to a given value. The attribute will be created if it does not exist, or changed if it does.

4.13.2.11 void TiXmlElement::SetAttribute (const char * *name*, int *value*)

Sets an attribute of name to a given value. The attribute will be created if it does not exist, or changed if it does.

4.13.2.12 void TiXmlElement::SetDoubleAttribute (const char * *name*, double *value*)

Sets an attribute of name to a given value. The attribute will be created if it does not exist, or changed if it does.

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

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp
- /home/cshome/m/mabrams/345/txtEngine/tinyxmlparser.cpp

4.14 TiXmlHandle Class Reference

`#include <tinyxml.h>`

Public Member Functions

- TiXmlHandle (TiXmlNode *_node)

  Create a handle from any node (at any depth of the tree.) This can be a null pointer.
- TiXmlHandle (const TiXmlHandle &ref)

  Copy constructor.
- TiXmlHandle **operator=** (const TiXmlHandle &ref)
- TiXmlHandle FirstChild () const

  Return a handle to the first child node.
- TiXmlHandle FirstChild (const char *value) const

  Return a handle to the first child node with the given name.
- TiXmlHandle FirstChildElement () const

  Return a handle to the first child element.
- TiXmlHandle FirstChildElement (const char *value) const

  Return a handle to the first child element with the given name.
- TiXmlHandle Child (const char *value, int index) const
- TiXmlHandle Child (int index) const
- TiXmlHandle ChildElement (const char *value, int index) const
- TiXmlHandle ChildElement (int index) const
- TiXmlNode * ToNode () const
- TiXmlElement * ToElement () const
- TiXmlText * ToText () const
- TiXmlUnknown * ToUnknown () const
- TiXmlNode * Node () const

- TiXmlElement * Element () const
- TiXmlText * Text () const
- TiXmlUnknown * Unknown () const

4.14.1 Detailed Description

A TiXmlHandle is a class that wraps a node pointer with null checks; this is an incredibly useful thing. Note that TiXmlHandle is not part of the TinyXml DOM structure. It is a separate utility class.

Take an example:

```
<Document>
<Element attributeA = "valueA">
<Child attributeB = "value1" />
<Child attributeB = "value2" />
</Element>
<Document>
```

Assuming you want the value of "attributeB" in the 2nd "Child" element, it's very easy to write a *lot* of code that looks like:

```
TiXmlElement* root = document.FirstChildElement( "Document" );
if ( root )
{
TiXmlElement* element = root->FirstChildElement( "Element" );
if ( element )
{
TiXmlElement* child = element->FirstChildElement( "Child" );
if ( child )
{
TiXmlElement* child2 = child->NextSiblingElement( "Child" );
if ( child2 )
{
// Finally do something useful.
```

And that doesn't even cover "else" cases. TiXmlHandle addresses the verbosity of such code. A TiXmlHandle checks for null pointers so it is perfectly safe and correct to use:

```
TiXmlHandle docHandle( &document );
TiXmlElement* child2 = docHandle.FirstChild( "Document" ).FirstChild( "Element" ).Child( "Chi
if ( child2 )
{
// do something useful
```

Which is MUCH more concise and useful.

It is also safe to copy handles - internally they are nothing more than node pointers.

```
TiXmlHandle handleCopy = handle;
```

What they should not be used for is iteration:

```
int i=0;
while ( true )
{
TiXmlElement* child = docHandle.FirstChild( "Document" ).FirstChild( "Element" ).Child( "Child", i ).ToE
if ( !child )
break;
// do something
++i;
}
```

It seems reasonable, but it is in fact two embedded while loops. The Child method is a linear walk to find the element, so this code would iterate much more than it needs to. Instead, prefer:

```
TiXmlElement* child = docHandle.FirstChild( "Document" ).FirstChild( "Element" ).FirstChild( "Child" ).T

for( child; child; child=child->NextSiblingElement() )
{
// do something
}
```

4.14.2 Member Function Documentation

4.14.2.1 TiXmlHandle TiXmlHandle::Child (const char * *value,* int *index*) const

Return a handle to the "index" child with the given name. The first child is 0, the second 1, etc.

4.14.2.2 TiXmlHandle TiXmlHandle::Child (int *index*) const

Return a handle to the "index" child. The first child is 0, the second 1, etc.

4.14.2.3 TiXmlHandle TiXmlHandle::ChildElement (const char * *value,* int *index*) const

Return a handle to the "index" child element with the given name. The first child element is 0, the second 1, etc. Note that only TiXmlElements are indexed: other types are not counted.

4.14.2.4 TiXmlHandle TiXmlHandle::ChildElement (int *index*) const

Return a handle to the "index" child element. The first child element is 0, the second 1, etc. Note that only TiXmlElements are indexed: other types are not counted.

4.14.2.5 TiXmlElement* TiXmlHandle::Element ( ) const [inline]

Deprecated

use ToElement. Return the handle as a TiXmlElement. This may return null.

4.14.2.6 TiXmlNode* TiXmlHandle::Node ( ) const [inline]

Deprecated

use ToNode. Return the handle as a TiXmlNode. This may return null.

4.14.2.7 TiXmlText* TiXmlHandle::Text ( ) const [inline]

Deprecated

use ToText() Return the handle as a TiXmlText. This may return null.

4.14.2.8 TiXmlElement* TiXmlHandle::ToElement ( ) const [inline]

Return the handle as a TiXmlElement. This may return null.

4.14.2.9 TiXmlNode* TiXmlHandle::ToNode ( ) const [inline]

Return the handle as a TiXmlNode. This may return null.

4.14.2.10 TiXmlText* TiXmlHandle::ToText ( ) const [inline]

Return the handle as a TiXmlText. This may return null.

4.14.2.11 TiXmlUnknown* TiXmlHandle::ToUnknown ( ) const [inline]

Return the handle as a TiXmlUnknown. This may return null.

4.14.2.12 TiXmlUnknown* TiXmlHandle::Unknown ( ) const [inline]

Deprecated

use ToUnknown() Return the handle as a TiXmlUnknown. This may return null.

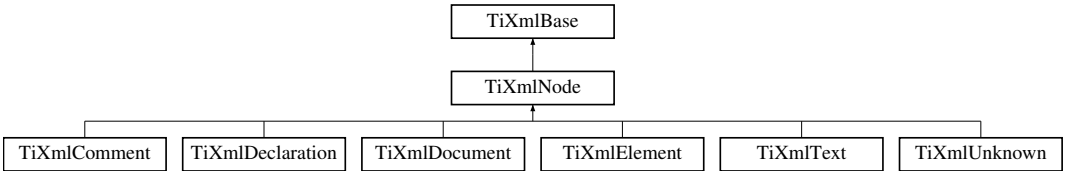
The documentation for this class was generated from the following files:

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp

4.15 TiXmlNode Class Reference

`#include <tinyxml.h>`

Inheritance diagram for TiXmlNode:



Public Types

- enum NodeType {

  TINYXML_DOCUMENT, **TINYXML_ELEMENT**, **TINYXML_COMMENT**, **TINYXML_-UNKNOWN**,

  TINYXML_TEXT, **TINYXML_DECLARATION**, **TINYXML_TYPECOUNT** }

Public Member Functions

- const char * Value () const
- const TIXML_STRING & **ValueTStr** () const
- void SetValue (const char *_value)
- void Clear ()

  Delete all the children of this node. Does not affect 'this'.
- TiXmlNode * Parent ()

  One step up the DOM.
- const TiXmlNode * **Parent** () const
- const TiXmlNode * FirstChild () const

  The first child of this node. Will be null if there are no children.
- TiXmlNode * **FirstChild** ()
- const TiXmlNode * FirstChild (const char *value) const
- TiXmlNode * FirstChild (const char *_value)

  The first child of this node with the matching 'value'. Will be null if none found.
- const TiXmlNode * **LastChild** () const
- TiXmlNode * LastChild ()

  The last child of this node. Will be null if there are no children.
- const TiXmlNode * **LastChild** (const char *value) const
- TiXmlNode * LastChild (const char *_value)

  The last child of this node matching 'value'. Will be null if there are no children.
- const TiXmlNode * IterateChildren (const TiXmlNode *previous) const
- TiXmlNode * **IterateChildren** (const TiXmlNode *previous)
- const TiXmlNode * IterateChildren (const char *value, const TiXmlNode *previous) const

  This flavor of IterateChildren searches for children with a particular 'value'.
- TiXmlNode * **IterateChildren** (const char *_value, const TiXmlNode *previous)

- TiXmlNode * InsertEndChild (const TiXmlNode &addThis)
- TiXmlNode * LinkEndChild (TiXmlNode *addThis)
- TiXmlNode * InsertBeforeChild (TiXmlNode *beforeThis, const TiXmlNode &addThis)
- TiXmlNode * InsertAfterChild (TiXmlNode *afterThis, const TiXmlNode &addThis)
- TiXmlNode * ReplaceChild (TiXmlNode *replaceThis, const TiXmlNode &withThis)
- bool RemoveChild (TiXmlNode *removeThis)

  Delete a child of this node.
- const TiXmlNode * PreviousSibling () const

  Navigate to a sibling node.
- TiXmlNode * **PreviousSibling** ()
- const TiXmlNode * PreviousSibling (const char *) const

  Navigate to a sibling node.
- TiXmlNode * **PreviousSibling** (const char *_prev)
- const TiXmlNode * NextSibling () const

  Navigate to a sibling node.
- TiXmlNode * **NextSibling** ()
- const TiXmlNode * NextSibling (const char *) const

  Navigate to a sibling node with the given 'value'.
- TiXmlNode * **NextSibling** (const char *_next)
- const TiXmlElement * NextSiblingElement () const
- TiXmlElement * **NextSiblingElement** ()
- const TiXmlElement * NextSiblingElement (const char *) const
- TiXmlElement * **NextSiblingElement** (const char *_next)
- const TiXmlElement * FirstChildElement () const

  Convenience function to get through elements.
- TiXmlElement * **FirstChildElement** ()
- const TiXmlElement * FirstChildElement (const char *_value) const

  Convenience function to get through elements.
- TiXmlElement * **FirstChildElement** (const char *_value)
- int Type () const
- const TiXmlDocument * GetDocument () const
- TiXmlDocument * **GetDocument** ()
- bool NoChildren () const

  Returns true if this node has no children.
- virtual const TiXmlDocument * ToDocument () const

  Cast to a more defined type. Will return null if not of the requested type.
- virtual const TiXmlElement * ToElement () const

  Cast to a more defined type. Will return null if not of the requested type.
- virtual const TiXmlComment * ToComment () const

  Cast to a more defined type. Will return null if not of the requested type.
- virtual const TiXmlUnknown * ToUnknown () const

  Cast to a more defined type. Will return null if not of the requested type.
- virtual const TiXmlText * ToText () const

Cast to a more defined type. Will return null if not of the requested type.

- virtual const TiXmlDeclaration * ToDeclaration () const

  Cast to a more defined type. Will return null if not of the requested type.
- virtual TiXmlDocument * ToDocument ()

  Cast to a more defined type. Will return null if not of the requested type.
- virtual TiXmlElement * ToElement ()

  Cast to a more defined type. Will return null if not of the requested type.
- virtual TiXmlComment * ToComment ()

  Cast to a more defined type. Will return null if not of the requested type.
- virtual TiXmlUnknown * ToUnknown ()

  Cast to a more defined type. Will return null if not of the requested type.
- virtual TiXmlText * ToText ()

  Cast to a more defined type. Will return null if not of the requested type.
- virtual TiXmlDeclaration * ToDeclaration ()

  Cast to a more defined type. Will return null if not of the requested type.
- virtual TiXmlNode * Clone () const =0
- virtual bool Accept (TiXmlVisitor *visitor) const =0

Protected Member Functions

- **TiXmlNode** (NodeType _type)
- void **CopyTo** (TiXmlNode *target) const
- TiXmlNode * **Identify** (const char *start, TiXmlEncoding encoding)

Protected Attributes

- TiXmlNode * **parent**
- NodeType **type**
- TiXmlNode * **firstChild**
- TiXmlNode * **lastChild**
- TIXML_STRING **value**
- TiXmlNode * **prev**
- TiXmlNode * **next**

Friends

- class TiXmlDocument
- class TiXmlElement

4.15.1 Detailed Description

The parent class for everything in the Document Object Model. (Except for attributes). Nodes have siblings, a parent, and children. A node can be in a document, or stand on its own. The type of a TiXmlNode can be queried, and it can be cast to its more defined type.

4.15.2 Member Enumeration Documentation

4.15.2.1 enum TiXmlNode::NodeType

The types of XML nodes supported by TinyXml. (All the unsupported types are picked up by UNKNOWN.)

4.15.3 Member Function Documentation

4.15.3.1 virtual bool TiXmlNode::Accept (TiXmlVisitor * *visitor*) const [pure virtual]

Accept a hierchical visit the nodes in the TinyXML DOM. Every node in the XML tree will be conditionally visited and the host will be called back via the TiXmlVisitor interface.

This is essentially a SAX interface for TinyXML. (Note however it doesn't re-parse the XML for the callbacks, so the performance of TinyXML is unchanged by using this interface versus any other.)

The interface has been based on ideas from:

- http://www.saxproject.org/
- http://c2.com/cgi/wiki?HierarchicalVisitorPattern

Which are both good references for "visiting".

An example of using Accept():

```
TiXmlPrinter printer;
tinyxmlDoc.Accept( &printer );
const char* xmlcstr = printer.CStr();
```

Implemented in TiXmlElement, TiXmlComment, TiXmlText, TiXmlDeclaration, TiXmlUnknown, and TiXmlDocument.

4.15.3.2 virtual TiXmlNode* TiXmlNode::Clone ( ) const [pure virtual]

Create an exact duplicate of this node and return it. The memory must be deleted by the caller.

Implemented in TiXmlElement, TiXmlComment, TiXmlText, TiXmlDeclaration, TiXmlUnknown, and TiXmlDocument.

4.15.3.3 const TiXmlNode * TiXmlNode::FirstChild (const char * *value*) const

The first child of this node with the matching 'value'. Will be null if none found.

4.15.3.4 const TiXmlDocument * TiXmlNode::GetDocument ( ) const

Return a pointer to the Document this node lives in. Returns null if not in a document.

4.15.3.5 TiXmlNode * TiXmlNode::InsertAfterChild (TiXmlNode * *afterThis,* const TiXmlNode & *addThis*)

Add a new node related to this. Adds a child after the specified child. Returns a pointer to the new object or NULL if an error occured.

4.15.3.6 TiXmlNode * TiXmlNode::InsertBeforeChild (TiXmlNode * *beforeThis,* const TiXmlNode & *addThis*)

Add a new node related to this. Adds a child before the specified child. Returns a pointer to the new object or NULL if an error occured.

4.15.3.7 TiXmlNode * TiXmlNode::InsertEndChild (const TiXmlNode & *addThis*)

Add a new node related to this. Adds a child past the LastChild. Returns a pointer to the new object or NULL if an error occured.

4.15.3.8 const TiXmlNode * TiXmlNode::IterateChildren (const TiXmlNode * *previous*) const

An alternate way to walk the children of a node. One way to iterate over nodes is:

```
for( child = parent->FirstChild(); child; child = child->NextSibling() )
```

IterateChildren does the same thing with the syntax:

```
child = 0;
while( child = parent->IterateChildren( child ) )
```

IterateChildren takes the previous child as input and finds the next one. If the previous child is null, it returns the first. IterateChildren will return null when done.

4.15.3.9 TiXmlNode * TiXmlNode::LinkEndChild (TiXmlNode * *addThis*)

Add a new node related to this. Adds a child past the LastChild.

NOTE: the node to be added is passed by pointer, and will be henceforth owned (and deleted) by tinyXml. This method is efficient and avoids an extra copy, but should be used with care as it uses a different memory model than the other insert functions.

See also

InsertEndChild

4.15.3.10 const TiXmlElement * TiXmlNode::NextSiblingElement (const char * *_value*) const

Convenience function to get through elements. Calls NextSibling and ToElement. Will skip all non-Element nodes. Returns 0 if there is not another element.

4.15.3.11 const TiXmlElement * TiXmlNode::NextSiblingElement ( ) const

Convenience function to get through elements. Calls NextSibling and ToElement. Will skip all non-Element nodes. Returns 0 if there is not another element.

4.15.3.12 TiXmlNode * TiXmlNode::ReplaceChild (TiXmlNode * *replaceThis*, const TiXmlNode & *withThis*)

Replace a child of this node. Returns a pointer to the new object or NULL if an error occured.

4.15.3.13 void TiXmlNode::SetValue (const char * *_value*) [inline]

Changes the value of the node. Defined as:

```
Document: filename of the xml file
Element: name of the element
Comment: the comment text
Unknown: the tag contents
Text: the text string
```

4.15.3.14 int TiXmlNode::Type ( ) const [inline]

Query the type (as an enumerated value, above) of this node. The possible types are: TINYXML_DOCUMENT, TINYXML_ELEMENT, TINYXML_COMMENT, TINYXML_UNKNOWN, TINYXML_TEXT, and TINYXML_DECLARATION.

4.15.3.15 const char* TiXmlNode::Value ( ) const [inline]

The meaning of 'value' changes for the specific type of TiXmlNode.

```
Document: filename of the xml file
Element: name of the element
Comment: the comment text
Unknown: the tag contents
Text: the text string
```

The subclasses will wrap this function.

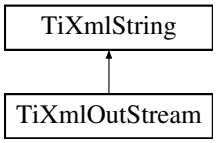
The documentation for this class was generated from the following files:

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp
- /home/cshome/m/mabrams/345/txtEngine/tinyxmlparser.cpp

4.16 TiXmlOutStream Class Reference

Inheritance diagram for TiXmlOutStream:



Public Member Functions

- TiXmlOutStream & **operator**<< (const TiXmlString &in)
- TiXmlOutStream & **operator**<< (const char *in)

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

- /home/cshome/m/mabrams/345/txtEngine/tinystr.h

4.17 TiXmlParsingData Class Reference

Public Member Functions

- void **Stamp** (const char *now, TiXmlEncoding encoding)
- const TiXmlCursor & **Cursor** () const

Friends

- class TiXmlDocument

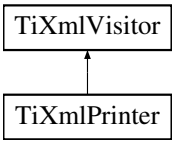
The documentation for this class was generated from the following file:

- /home/cshome/m/mabrams/345/txtEngine/tinyxmlparser.cpp

4.18 TiXmlPrinter Class Reference

`#include <tinyxml.h>`

Inheritance diagram for TiXmlPrinter:



Public Member Functions

- virtual bool VisitEnter (const TiXmlDocument &doc)

  Visit a document.
- virtual bool VisitExit (const TiXmlDocument &doc)

  Visit a document.
- virtual bool VisitEnter (const TiXmlElement &element, const TiXmlAttribute *firstAttribute)

  Visit an element.
- virtual bool VisitExit (const TiXmlElement &element)

  Visit an element.
- virtual bool Visit (const TiXmlDeclaration &declaration)

  Visit a declaration.
- virtual bool Visit (const TiXmlText &text)

  Visit a text node.
- virtual bool Visit (const TiXmlComment &comment)

  Visit a comment node.
- virtual bool Visit (const TiXmlUnknown &unknown)

  Visit an unknown node.
- void SetIndent (const char *_indent)
- const char * Indent ()

  Query the indention string.
- void SetLineBreak (const char *_lineBreak)
- const char * LineBreak ()

  Query the current line breaking string.
- void SetStreamPrinting ()
- const char * CStr ()

  Return the result.
- size_t Size ()

  Return the length of the result string.

4.18.1 Detailed Description

Print to memory functionality. The TiXmlPrinter is useful when you need to:

1. Print to memory (especially in non-STL mode)

2. Control formatting (line endings, etc.)

When constructed, the TiXmlPrinter is in its default "pretty printing" mode. Before calling Accept() you can call methods to control the printing of the XML document. After TiXmlNode::Accept() is called, the printed document can be accessed via the CStr(), Str(), and Size() methods.

TiXmlPrinter uses the Visitor API.

```
TiXmlPrinter printer;
printer.SetIndent( "\t" );

doc.Accept( &printer );
fprintf( stdout, "%s", printer.CStr() );
```

4.18.2 Member Function Documentation

4.18.2.1 void TiXmlPrinter::SetIndent (const char * _indent) [inline]

Set the indent characters for printing. By default 4 spaces but tab () is also useful, or null/empty string for no indentation.

4.18.2.2 void TiXmlPrinter::SetLineBreak (const char * _lineBreak) [inline]

Set the line breaking string. By default set to newline (

). Some operating systems prefer other characters, or can be set to the null/empty string for no indenation.

4.18.2.3 void TiXmlPrinter::SetStreamPrinting ( ) [inline]

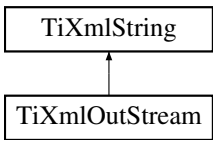
Switch over to "stream printing" which is the most dense formatting without linebreaks. Common when the XML is needed for network transmission.

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

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp

4.19 TiXmlString Class Reference

Inheritance diagram for TiXmlString:



Classes

- struct **Rep**

Public Types

- typedef size_t **size_type**

Public Member Functions

- **TiXmlString** (const TiXmlString &copy)
- TIXML_EXPLICIT **TiXmlString** (const char *copy)
- TIXML_EXPLICIT **TiXmlString** (const char *str, size_type len)
- TiXmlString & **operator=** (const char *copy)
- TiXmlString & **operator=** (const TiXmlString &copy)
- TiXmlString & **operator+=** (const char *suffix)
- TiXmlString & **operator+=** (char single)
- TiXmlString & **operator+=** (const TiXmlString &suffix)
- const char * **c_str** () const
- const char * **data** () const
- size_type **length** () const
- size_type **size** () const
- bool **empty** () const
- size_type **capacity** () const
- const char & **at** (size_type index) const
- char & **operator[]** (size_type index) const
- size_type **find** (char lookup) const
- size_type **find** (char tofind, size_type offset) const
- void **clear** ()
- void **reserve** (size_type cap)
- TiXmlString & **assign** (const char *str, size_type len)
- TiXmlString & **append** (const char *str, size_type len)
- void **swap** (TiXmlString &other)

Static Public Attributes

- static const size_type **npos** = static_cast< TiXmlString::size_type >(-1)

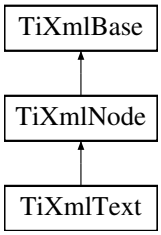
The documentation for this class was generated from the following files:

- /home/cshome/m/mabrams/345/txtEngine/tinystr.h
- /home/cshome/m/mabrams/345/txtEngine/tinystr.cpp

4.20 TiXmlText Class Reference

`#include <tinyxml.h>`

Inheritance diagram for TiXmlText:



Public Member Functions

- TiXmlText (const char ∗initValue)
- **TiXmlText** (const TiXmlText &copy)
- TiXmlText & **operator=** (const TiXmlText &base)
- virtual void Print (FILE ∗cfile, int depth) const
- bool CDATA () const

  Queries whether this represents text using a CDATA section.
- void SetCDATA (bool _cdata)

  Turns on or off a CDATA representation of text.
- virtual const char ∗ **Parse** (const char ∗p, TiXmlParsingData ∗data, TiXmlEncoding encoding)
- virtual const TiXmlText ∗ ToText () const

  Cast to a more defined type. Will return null not of the requested type.
- virtual TiXmlText ∗ ToText ()

  Cast to a more defined type. Will return null not of the requested type.
- virtual bool Accept (TiXmlVisitor ∗content) const

Protected Member Functions

- virtual TiXmlNode ∗ Clone () const

  [internal use] Creates a new Element and returns it.
- void **CopyTo** (TiXmlText ∗target) const
- bool **Blank** () const

Friends

- class TiXmlElement

4.20.1 Detailed Description

XML text. A text node can have 2 ways to output the next. "normal" output and CDATA. It will default to the mode it was parsed from the XML file and you generally want to leave it alone, but you can change the output mode with SetCDATA() and query it with CDATA().

4.20.2 Constructor & Destructor Documentation

4.20.2.1 TiXmlText::TiXmlText (const char * *initValue*) [inline]

Constructor for text element. By default, it is treated as normal, encoded text. If you want it be output as a CDATA text element, set the parameter _cdata to 'true'

4.20.3 Member Function Documentation

4.20.3.1 bool TiXmlText::Accept (TiXmlVisitor * *content*) const [virtual]

Walk the XML tree visiting this node and all of its children.

Implements TiXmlNode.

4.20.3.2 void TiXmlText::Print (FILE * *cfile,* int *depth*) const [virtual]

All TinyXml classes can print themselves to a filestream or the string class (TiXmlString in non-STL mode, std::string in STL mode.) Either or both cfile and str can be null.

This is a formatted print, and will insert tabs and newlines.

(For an unformatted stream, use the << operator.)

Implements TiXmlBase.

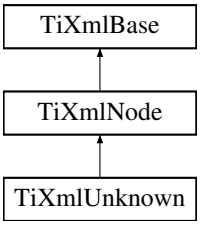
The documentation for this class was generated from the following files:

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp
- /home/cshome/m/mabrams/345/txtEngine/tinyxmlparser.cpp

4.21 TiXmlUnknown Class Reference

`#include <tinyxml.h>`

Inheritance diagram for TiXmlUnknown:



Public Member Functions

- **TiXmlUnknown** (const TiXmlUnknown &copy)
- TiXmlUnknown & **operator=** (const TiXmlUnknown &copy)
- virtual TiXmlNode * Clone () const

  Creates a copy of this Unknown and returns it.
- virtual void Print (FILE *cfile, int depth) const
- virtual const char * **Parse** (const char *p, TiXmlParsingData *data, TiXmlEncoding encoding)
- virtual const TiXmlUnknown * ToUnknown () const

  Cast to a more defined type. Will return null not of the requested type.
- virtual TiXmlUnknown * ToUnknown ()

  Cast to a more defined type. Will return null not of the requested type.
- virtual bool Accept (TiXmlVisitor *content) const

Protected Member Functions

- void **CopyTo** (TiXmlUnknown *target) const

4.21.1 Detailed Description

Any tag that tinyXml doesn't recognize is saved as an unknown. It is a tag of text, but should not be modified. It will be written back to the XML, unchanged, when the file is saved.

DTD tags get thrown into TiXmlUnknowns.

4.21.2 Member Function Documentation

4.21.2.1 bool TiXmlUnknown::Accept (TiXmlVisitor * *content*) const [virtual]

Walk the XML tree visiting this node and all of its children.

Implements TiXmlNode.

4.21.2.2 void TiXmlUnknown::Print (FILE * *cfile*, int *depth*) const [virtual]

All TinyXml classes can print themselves to a filestream or the string class (TiXmlString in non-STL mode, std::string in STL mode.) Either or both cfile and str can be null.

This is a formatted print, and will insert tabs and newlines.

(For an unformatted stream, use the << operator.)

Implements TiXmlBase.

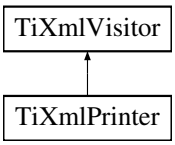
The documentation for this class was generated from the following files:

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h
- /home/cshome/m/mabrams/345/txtEngine/tinyxml.cpp
- /home/cshome/m/mabrams/345/txtEngine/tinyxmlparser.cpp

4.22 TiXmlVisitor Class Reference

`#include <tinyxml.h>`

Inheritance diagram for TiXmlVisitor:



Public Member Functions

- virtual bool VisitEnter (const TiXmlDocument &)
  Visit a document.
- virtual bool VisitExit (const TiXmlDocument &)
  Visit a document.
- virtual bool VisitEnter (const TiXmlElement &, const TiXmlAttribute *)
  Visit an element.
- virtual bool VisitExit (const TiXmlElement &)
  Visit an element.
- virtual bool Visit (const TiXmlDeclaration &)
  Visit a declaration.
- virtual bool Visit (const TiXmlText &)
  Visit a text node.
- virtual bool Visit (const TiXmlComment &)
  Visit a comment node.
- virtual bool Visit (const TiXmlUnknown &)
  Visit an unknown node.

4.22.1 Detailed Description

Implements the interface to the "Visitor pattern" (see the Accept() method.) If you call the Accept() method, it requires being passed a TiXmlVisitor class to handle callbacks. For nodes that contain other nodes (Document, Element) you will get called with a VisitEnter/VisitExit pair. Nodes that are always leaves are simply called with Visit().

If you return 'true' from a Visit method, recursive parsing will continue. If you return false, **no children of this node or its sibilings** will be Visited.

All flavors of Visit methods have a default implementation that returns 'true' (continue visiting). You need to only override methods that are interesting to you.

Generally Accept() is called on the TiXmlDocument, although all nodes suppert Visiting.

You should never change the document from a callback.

See also

TiXmlNode::Accept()

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

- /home/cshome/m/mabrams/345/txtEngine/tinyxml.h

4.23 World Class Reference

Public Member Functions

- std::string **get_author** ()
- std::string **get_language** ()
- Area * **get_active_area** ()
- Area * **get_area** (int index)
- void **add_area** (Area *new_area)
- int **get_num_areas** ()
- Area * **get_area** (std::string area_id)
- bool **init_active_area** ()
- void **change_area** (std::string name)
- void **command** (std::string command)
- **World** (const char *lang, const char *auth, const char *init_area)

Protected Attributes

- std::string **language**
- std::string **author**
- std::vector< Area * > **areas**
- std::string **initial_area**
- int **num_areas**

- Area * **active_area**

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

- /home/cshome/m/mabrams/345/txtEngine/World.h

Index