

Clippings Libray

0.1

Generated by Doxygen 1.7.3

Tue Aug 30 2011 10:45:33

Contents

1	Todo List	1
2	Namespace Documentation	1
2.1	Package WT	1
2.1.1	Detailed Description	1
2.2	Package WT.Kindle	1
2.2.1	Detailed Description	2
2.3	Package WT.Kindle.Clippings	2
2.3.1	Detailed Description	2
3	Class Documentation	2
3.1	WT.Kindle.Clippings.Book Class Reference	2
3.1.1	Detailed Description	4
3.1.2	Constructor & Destructor Documentation	4
3.1.3	Member Function Documentation	5
3.1.4	Member Data Documentation	9
3.2	WT.Kindle.Clippings.Clipping Class Reference	10
3.2.1	Detailed Description	12
3.2.2	Constructor & Destructor Documentation	13
3.2.3	Member Function Documentation	14
3.2.4	Member Data Documentation	26
3.3	WT.Kindle.Clippings.Utility Class Reference	27
3.3.1	Detailed Description	28
3.3.2	Member Function Documentation	29
3.3.3	Member Data Documentation	38
4	File Documentation	38
4.1	Book.java File Reference	38
4.1.1	Detailed Description	39
4.2	Book.java	39
4.3	Clipping.java File Reference	40
4.3.1	Detailed Description	41
4.4	Clipping.java	41
4.5	Utility.java File Reference	44
4.5.1	Detailed Description	45
4.6	Utility.java	45

1 Todo List

Class `WT.Kindle.Clippings.Book` Add code to handle errors such as incorrect file layout.

Member `WT.Kindle.Clippings.Book.add(Clipping clipping)` Add error handling code

Member `WT.Kindle.Clippings.Book.setTitle(String title)` Add error handling code

Class [WT.Kindle.Clippings.Clipping](#) Add code to handle errors such as incorrect file layout.

Class [WT.Kindle.Clippings.Utility](#) Add code to handle errors such as incorrect file layout.

2 Namespace Documentation

2.1 Package WT

Package containing code from developerAntony Jordan.

Packages

- package [Kindle](#)
Package containing all code relating to the kindle.

2.1.1 Detailed Description

Package containing code from developerAntony Jordan.

2.2 Package WT.Kindle

Package containing all code relating to the kindle.

Packages

- package [Clippings](#)
Package containing the code for the ClippingsParser library.

2.2.1 Detailed Description

Package containing all code relating to the kindle.

2.3 Package WT.Kindle.Clippings

Package containing the code for the ClippingsParser library.

Classes

- class [Book](#)

The [Book](#) class stores all the clippings from a particular book title in a single location. It uses the [Clipping](#) class to store the clippings as an object in an `ArrayList` and stores their common title as a single string.

- class [Clipping](#)

The class stores the data for a single clipping from the "My Clippings.txt" of the kindle. Methods are provided to parse each item in the clipping into individual objects.

- class [Utility](#)

The class contains useful methods for the `ClippingsParser` library. Methods are provided to parse the "My Clippings.txt" file found on kindle devices into a useful format using the [Clipping](#) and [Book](#) classes.

2.3.1 Detailed Description

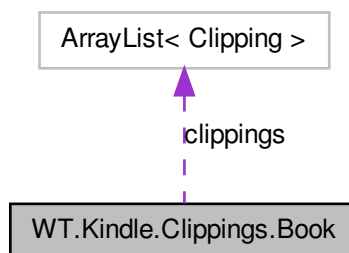
Package containing the code for the `ClippingsParser` library.

3 Class Documentation

3.1 WT.Kindle.Clippings.Book Class Reference

The [Book](#) class stores all the clippings from a particular book title in a single location. It uses the [Clipping](#) class to store the clippings as an object in an `ArrayList` and stores their common title as a single string.

Collaboration diagram for `WT.Kindle.Clippings.Book`:



Public Member Functions

- [Book](#) ()
This constructor initialises the title to an empty string and the clippings to an empty `ArrayList<Clipping>`.
- [Book](#) (String title)
This constructor initialises the title to a given string value and the clippings to an empty `ArrayList<Clipping>`.
- boolean [isTitleSet](#) ()
Used to check if the book's title has been set.
- void [add](#) ([Clipping](#) clipping)
Adds a single clipping to the book if its title matches the book's.
- void [setTitle](#) (String title)
Sets the common title for the clippings.
- void [setClippings](#) (`ArrayList< Clipping >` clippings)
Overwrites the current clippings `ArrayList` with the one provided.
- String [getTitle](#) ()
Get method to retrieve the common title for the clippings stored.
- `ArrayList< Clipping >` [getClippings](#) ()
Get method to retrieve the clippings stored.
- String [toXML](#) (int indentation)
Outputs the current book and all its clippings in XML format.
- String [toString](#) ()
Displays the book title along with the number of clippings shown in parenthesis.

Private Attributes

- String [title](#)
A string value to store the common title of all the clippings.
- `ArrayList< Clipping >` [clippings](#)
An `ArrayList` of [Clipping](#) to store all of the clippings common to the title.
- boolean [isTitleSet](#)
A boolean value used to check if the title has been set for the book.

3.1.1 Detailed Description

The [Book](#) class stores all the clippings from a particular book title in a single location. It uses the [Clipping](#) class to store the clippings as an object in an ArrayList and stores their common title as a single string.

Date

Created: 25/08/2011
Modified: 27/08/2011

Author

Antony Jordan

Todo

Add code to handle errors such as incorrect file layout.

Definition at line 27 of file [Book.java](#).

3.1.2 Constructor & Destructor Documentation

3.1.2.1 WT.Kindle.Clippings.Book.Book ()

This constructor initialises the title to an empty string and the clippings to an empty ArrayList<Clipping>.

Definition at line 36 of file [Book.java](#).

References [WT.Kindle.Clippings.Book.clippings](#), [WT.Kindle.Clippings.Book.isTitleSet](#), and [WT.Kindle.Clippings.Book.title](#).

```
    {  
        this.title = "";  
        clippings = new ArrayList<Clipping>();  
        isTitleSet = false;  
    }
```

3.1.2.2 WT.Kindle.Clippings.Book.Book (String title)

This constructor initialises the title to a given string value and the clippings to an empty ArrayList<Clipping>.

Parameters

<i>title</i>	The given string value for the title.
--------------	---------------------------------------

Definition at line 45 of file [Book.java](#).

References [WT.Kindle.Clippings.Book.clippings](#), [WT.Kindle.Clippings.Book.isTitleSet](#), and [WT.Kindle.Clippings.Book.title](#).

```
    {
        this.title = title;
        clippings = new ArrayList<Clipping>();
        isTitleSet = true;
    }
```

3.1.3 Member Function Documentation

3.1.3.1 boolean WT.Kindle.Clippings.Book.IsTitleSet ()

Used to check if the books title has been set.

Returns

True if it has been set False otherwise.

Definition at line 55 of file [Book.java](#).

References [WT.Kindle.Clippings.Book.isTitleSet](#).

```
{return isTitleSet;}
```

3.1.3.2 void WT.Kindle.Clippings.Book.add (Clipping *clipping*)

Adds a single clipping to the book if its title matches the book's.

Parameters

<i>clipping</i> The Clipping to add to the book.
--

Todo

Add error handling code

Definition at line 61 of file [Book.java](#).

References [WT.Kindle.Clippings.Book.clippings](#), [WT.Kindle.Clippings.Clipping.getTitle\(\)](#), and [WT.Kindle.Clippings.Book.title](#).

```
    {
        if(title.equals(clipping.getTitle())){
            clippings.add(clipping);
        }
    }
```

```
        else{  
        }  
    }
```

Here is the call graph for this function:



3.1.3.3 void WT.Kindle.Clippings.Book.setTitle (String title)

Sets the common title for the clippings.

Parameters

<i>title</i>	The String value to set as the title.
--------------	---------------------------------------

Todo

Add error handling code

Definition at line 74 of file [Book.java](#).

References [WT.Kindle.Clippings.Book.isTitleSet](#), and [WT.Kindle.Clippings.Book.title](#).

```
        {  
        if(!isTitleSet){  
            this.title = title;  
        }  
        else{  
        }  
    }
```

3.1.3.4 void WT.Kindle.Clippings.Book.setClippings (ArrayList< Clipping > clippings)

Overwrites the current clippings ArrayList with the one provided.

Parameters

<i>clippings</i> The ArrayList<Clipping> to replace the current one.
--

Definition at line 86 of file [Book.java](#).

References [WT.Kindle.Clippings.Book.clippings](#).

```
{this.clippings = clippings;}
```

3.1.3.5 String WT.Kindle.Clippings.Book.getTitle ()

Get method to retrieve the common title for the clippings stored.

Returns

The title common to all the clippings.

Definition at line 92 of file [Book.java](#).

References [WT.Kindle.Clippings.Book.title](#).

```
{return title;}
```

3.1.3.6 ArrayList<Clipping> WT.Kindle.Clippings.Book.getClippings ()

Get method to retrieve the clippings stored.

Returns

An ArrayList<Clipping> of all the stored clippings..

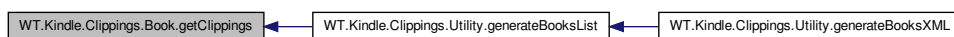
Definition at line 97 of file [Book.java](#).

References [WT.Kindle.Clippings.Book.clippings](#).

Referenced by [WT.Kindle.Clippings.Utility.generateBooksList\(\)](#).

```
{return clippings;}
```

Here is the caller graph for this function:



3.1.3.7 String WT.Kindle.Clippings.Book.toXML (int indentation)

Outputs the current book and all its clippings in XML format.

Parameters

<i>indentation</i>	The number of tabs to indent by.
--------------------	----------------------------------

Returns

A string containing the XML code.

Definition at line 104 of file [Book.java](#).

References [WT.Kindle.Clippings.Book.clippings](#), and [WT.Kindle.Clippings.Book.title](#).

```
        {
            String indent = "";
            for(int i = 0;i<indentation;i++){indent += "\t";}

            String xml = "";

            xml += "\n"      + indent + "<Book>";

            xml += "\n\t"   + indent + "<Title>" + title + "</Title>";

            for (Clipping clipping : clippings) {
                xml+=clipping.toXML(indentation + 1);
            }

            xml += "\n"      + indent + "</Book>";

            return xml;
        }
```

3.1.3.8 String WT.Kindle.Clippings.Book.toString ()

Displays the book title along with the number of clippings shown in parenthesis.

Definition at line 123 of file [Book.java](#).

References [WT.Kindle.Clippings.Book.clippings](#), and [WT.Kindle.Clippings.Book.title](#).

```
        {
            String retVal;

            retVal = title + "(" + clippings.size() + ")";

            return retVal;
        }
```

3.1.4 Member Data Documentation

3.1.4.1 String WT.Kindle.Clippings.Book.title [private]

A string value to store the common title of all the clippings.

Definition at line 29 of file [Book.java](#).

Referenced by [WT.Kindle.Clippings.Book.add\(\)](#), [WT.Kindle.Clippings.Book.Book\(\)](#), [WT.Kindle.Clippings.Book.getTitle\(\)](#), [WT.Kindle.Clippings.Book.setTitle\(\)](#), [WT.Kindle.Clippings.Book.toString\(\)](#), and [WT.Kindle.Clippings.Book.toXML\(\)](#).

3.1.4.2 ArrayList<Clipping> WT.Kindle.Clippings.Book.clippings [private]

An ArrayList of [Clipping](#) to store all of the clippings common to the title.

Definition at line 31 of file [Book.java](#).

Referenced by [WT.Kindle.Clippings.Book.add\(\)](#), [WT.Kindle.Clippings.Book.Book\(\)](#), [WT.Kindle.Clippings.Book.getClippings\(\)](#), [WT.Kindle.Clippings.Book.setClippings\(\)](#), [WT.Kindle.Clippings.Book.toString\(\)](#), and [WT.Kindle.Clippings.Book.toXML\(\)](#).

3.1.4.3 boolean WT.Kindle.Clippings.Book.isTitleSet [private]

A boolean value used to check if the title has been set for the book.

Definition at line 33 of file [Book.java](#).

Referenced by [WT.Kindle.Clippings.Book.Book\(\)](#), [WT.Kindle.Clippings.Book.IsTitleSet\(\)](#), and [WT.Kindle.Clippings.Book.setTitle\(\)](#).

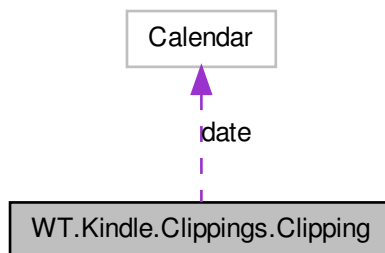
The documentation for this class was generated from the following file:

- [Book.java](#)

3.2 WT.Kindle.Clippings.Clipping Class Reference

The class stores the data for a single clipping from the "My Clippings.txt" of the kindle. Methods are provided to parse each item in the clipping into individual objects.

Collaboration diagram for WT.Kindle.Clippings.Clipping:



Public Member Functions

- [Clipping \(\)](#)
Constructor that initialises class to null values.
- [Clipping \(String input\)](#)
Constructor that takes a clipping as a string and parses each individual element into the correct fields.
- String [getAuthor \(\)](#)
Gets the author of the CLipping.
- String [getTitle \(\)](#)
Gets the title of the CLipping.
- int[] [getStartEnd \(\)](#)
Gets the start and end locations of the CLipping.
- String [getType \(\)](#)
Gets the type of the CLipping.
- Calendar [getDate \(\)](#)
Gets the date and time of the CLipping.
- String [getText \(\)](#)
Gets the text of the CLipping.
- void [setAuthor \(String author\)](#)
Sets the author of the clipping.

- void [setTitle](#) (String [title](#))
Sets the title of the clipping.
- void [setStartEnd](#) (int[] [startEnd](#))
Sets the start and end location of the clipping.
- void [setType](#) (String [type](#))
Sets the type of the clipping.
- void [setDate](#) (Calendar [date](#))
Sets the date of the clipping.
- void [setText](#) (String [text](#))
Sets the text of the clipping.
- String [toXML](#) (int indentation)
Outputs the current clipping in XML format.
- String [toString](#) ()
Gives a string representation of the [Clipping](#).

Private Member Functions

- String [parseAuthor](#) (String input)
Parses the author of the clipping from the input string.
- String [parseTitle](#) (String input)
Parses the title of the clipping from the input string.
- int[] [parseStartEnd](#) (String input)
Parses the start and end locations of the clipping from the input string.
- String [parseType](#) (String input)
Parses the type of the clipping from the input string.
- Calendar [parseDateTime](#) (String input)
Parses the date and time of the clipping from the input string.
- String [parseText](#) (String input)
Parses the text of the clipping from the input string. That is the highlighted text or user inputed note.

Private Attributes

- String [author](#)
String to store the author of the clipping.
- String [title](#)
String to store the title of the clipping.
- int[] [startEnd](#)
Integer array to hold the start and end location of the clipping.
- String [type](#)
String to hold the type of the clipping.
- Calendar [date](#)
Calendar to store the date and time of the clipping.
- String [text](#)
String to hold the text of the clipping. Either the highlighted text or the user's note.

3.2.1 Detailed Description

The class stores the data for a single clipping from the "My Clippings.txt" of the kindle. Methods are provided to parse each item in the clipping into individual objects.

Date

Created: 25/08/2011
Modified: 27/08/2011

Author

Antony Jordan

Todo

Add code to handle errors such as incorrect file layout.

Definition at line 30 of file [Clipping.java](#).

3.2.2 Constructor & Destructor Documentation

3.2.2.1 WT.Kindle.Clippings.Clipping.Clipping ()

Constructor that initialises class to null values.

Definition at line 46 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.author](#), [WT.Kindle.Clippings.Clipping.date](#), [WT.Kindle.Clippings.Clipping.startEnd](#), [WT.Kindle.Clippings.Clipping.text](#), [WT.Kindle.Clippings.Clipping.title](#), and [WT.Kindle.Clippings.Clipping.type](#).

```
    {  
        author = null;  
        title = null;  
        startEnd = null;  
        type = null;  
        date = Calendar.getInstance();  
        text = null;  
    }
```

3.2.2.2 WT.Kindle.Clippings.Clipping.Clipping (String *input*)

Constructor that takes a clipping as a string and parses each individual element into the correct fields.

Parameters

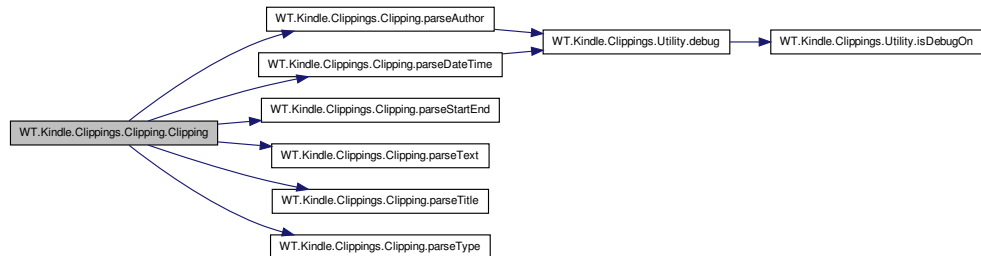
<i>input</i>	The clipping as a string.
--------------	---------------------------

Definition at line 58 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.author](#), [WT.Kindle.Clippings.Clipping.date](#), [WT.Kindle.Clippings.Clipping.parseAuthor\(\)](#), [WT.Kindle.Clippings.Clipping.parseDateTime\(\)](#), [WT.Kindle.Clippings.Clipping.parseStartEnd\(\)](#), [WT.Kindle.Clippings.Clipping.parseText\(\)](#), [WT.Kindle.Clippings.Clipping.parseTitle\(\)](#), [WT.Kindle.Clippings.Clipping.parseType\(\)](#), [WT.Kindle.Clippings.Clipping.startEnd](#), [WT.Kindle.Clippings.Clipping.text](#), [WT.Kindle.Clippings.Clipping.title](#), and [WT.Kindle.Clippings.Clipping.type](#).

```
    {  
        author = parseAuthor(input);  
        title = parseTitle(input);  
        startEnd = parseStartEnd(input);  
        type = parseType(input);  
        date = Calendar.getInstance();  
        date = parseDateTime(input);  
        text = parseText(input);  
    }
```

Here is the call graph for this function:



3.2.3 Member Function Documentation

3.2.3.1 String WT.Kindle.Clippings.Clipping.parseAuthor (String *input*) [private]

Parses the author of the clipping from the input string.

Parameters

<i>input</i> The clipping as a string.
--

Returns

A string holding just the author of the clipping.

Definition at line 73 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.author](#), and [WT.Kindle.Clippings.Utility.debug\(\)](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#).

```

        {
    String temp = input.substring(0, input.indexOf("\n"));

    Utility.debug("Clipping", "parseAuthor", "temp",temp);

    Integer authorStartIndex = temp.lastIndexOf("(");

    Utility.debug("Clipping", "parseAuthor", "authorStartIndex",authorStartIndex.toString());

    authorStartIndex++;

    Utility.debug("Clipping", "parseAuthor", "authorStartIndex",authorStartIndex.toString());

    Integer authorEndIndex = (temp.lastIndexOf(")"));
  
```



```

    Utility.debug("Clipping", "parseAuthor", "authorEndIndex",authorEndIndex.
toString());

    String author = temp.substring(authorStartIndex, authorEndIndex);

    author = author.trim();

    return author;
}

```

Here is the call graph for this function:



Here is the caller graph for this function:



3.2.3.2 String WT.Kindle.Clippings.Clipping.parseTitle (String *input*) [private]

Parses the title of the clipping from the input string.

Parameters

<i>input</i>	The clipping as a string.
--------------	---------------------------

Returns

A string holding just the title of the clipping.

Definition at line 101 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.title](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#).

{

```

String temp = input.substring(0, input.indexOf("\n"));
int authorStartIndex = temp.lastIndexOf("(");

String title = temp.substring(0, authorStartIndex);

title = title.trim();

return title;
}

```

Here is the caller graph for this function:



3.2.3.3 int [] WT.Kindle.Clippings.Clipping.parseStartEnd (String input) [private]

Parses the start and end locations of the clipping from the input string.

Parameters

<i>input</i>	The clipping as a string.
--------------	---------------------------

Returns

An array containing two values the first of which is the start location the last is the end location.

Definition at line 116 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.startEnd](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#).

```

{
int[] retVal = new int[2];
String startEnd = input.substring(input.indexOf("Loc.") + 5, input.indexOf(
"|"));
startEnd = startEnd.trim();
if (startEnd.contains("-")) {
String[] temp = startEnd.split("-");

temp[0] = temp[0].trim();
temp[1] = temp[1].trim();
}
}

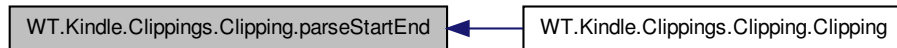
```

```

        retVal[0] = Integer.parseInt(temp[0]);
        retVal[1] = Integer.parseInt(temp[1]);
    }
    else
    {
        retVal[0] = retVal[1] = Integer.parseInt(startEnd);
    }
    return retVal;
}

```

Here is the caller graph for this function:



3.2.3.4 String WT.Kindle.Clippings.Clipping.parseType (String *input*) [private]

Parses the type of the clipping from the input string.

Parameters

input The clipping as a string.

Returns

A string holding just the type of the clipping.

Definition at line 140 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.type](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#).

```

        {
            String type = input.substring(input.indexOf("\n") + 3, input.indexOf("Loc.
"));
            type = type.trim();
            return type;
        }
    }

```

Here is the caller graph for this function:



3.2.3.5 Calendar WT.Kindle.Clippings.Clipping.parseDateTime (String *input*) [private]

Parses the date and time of the clipping from the input string.

Parameters

<i>input</i>	The clipping as a string.
--------------	---------------------------

Returns

A calendar representation of the date and time.

Definition at line 152 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.date](#), and [WT.Kindle.Clippings.Utility.debug\(\)](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#).

```

        {
String date = input.substring(input.indexOf("|"));

Utility.debug("Clipping", "parseDateTime", "date", date.replace("\n", "#"
));

if(date.contains("\n")){
    date = date.substring(0,date.indexOf("\n"));
}
date = date.replace("| Added on ", "");

DateFormat formatter = new SimpleDateFormat("EEEE, MMMM dd, yyyy, hh:mm a
");

Calendar calendar = Calendar.getInstance();
try {
    calendar.setTime(formatter.parse(date));
}
catch (ParseException e) {
    e.printStackTrace();
}

return calendar;
}
  
```

Here is the call graph for this function:



Here is the caller graph for this function:



3.2.3.6 String WT.Kindle.Clippings.Clipping.parseText (String *input*) [private]

Parses the text of the clipping from the input string. That is the highlighted text or user inputed note.

Parameters

<i>input</i>	The clipping as a string.
--------------	---------------------------

Returns

A string holding just the text of the clipping.

Definition at line 179 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.text](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#).

```

    {
        String text = "";
        if(text.contains("\n")){text = input.substring((input.indexOf("\n", input
        .indexOf("\n") + 1)));}
        text = text.trim();

        return text;
    }
  
```

Here is the caller graph for this function:



3.2.3.7 String WT.Kindle.Clippings.Clipping.getAuthor ()

Gets the author of the `Clipping`.

Returns

A string holding the author of the `Clipping`.

Definition at line 191 of file `Clipping.java`.

References `WT.Kindle.Clippings.Clipping.author`.

```
{return author;}
```

3.2.3.8 String WT.Kindle.Clippings.Clipping.getTitle ()

Gets the title of the `Clipping`.

Returns

A string holding the title of the `Clipping`.

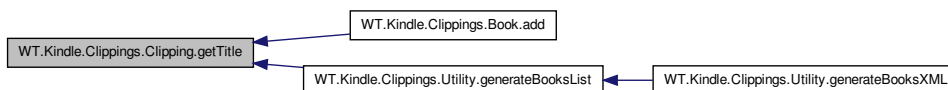
Definition at line 196 of file `Clipping.java`.

References `WT.Kindle.Clippings.Clipping.title`.

Referenced by `WT.Kindle.Clippings.Book.add()`, and `WT.Kindle.Clippings.Utility.generateBooksList()`.

```
{return title;}
```

Here is the caller graph for this function:



3.2.3.9 `int [] WT.Kindle.Clippings.Clipping.getStartEnd ()`

Gets the start and end locations of the CLipping.

Returns

An array containing two values the first of which is the start location the last is the end location.

Definition at line 201 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.startEnd](#).

```
{return startEnd; }
```

3.2.3.10 `String WT.Kindle.Clippings.Clipping.getType ()`

Gets the type of the CLipping.

Returns

A string holding the type of the [Clipping](#).

Definition at line 206 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.type](#).

```
{return type; }
```

3.2.3.11 `Calendar WT.Kindle.Clippings.Clipping.getDate ()`

Gets the date and time of the CLipping.

Returns

A calendar representing the date and time of the [Clipping](#).

Definition at line 211 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.date](#).

```
{return date;}
```

3.2.3.12 String WT.Kindle.Clippings.Clipping.getText ()

Gets the text of the [Clipping](#).

Returns

A string holding the text of the [Clipping](#).

Definition at line 216 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.text](#).

```
{return text;}
```

3.2.3.13 void WT.Kindle.Clippings.Clipping.setAuthor (String author)

Sets the author of the clipping.

Parameters

<i>author</i>	The value to set.
---------------	-------------------

Definition at line 222 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.author](#).

```
{this.author = author;}
```

3.2.3.14 void WT.Kindle.Clippings.Clipping.setTitle (String title)

Sets the title of the clipping.

Parameters

<i>title</i> The value to set.

Definition at line 227 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.title](#).

```
{this.title = title;}
```

3.2.3.15 void WT.Kindle.Clippings.Clipping.setStartEnd (int[] startEnd)

Sets the start and end location of the clipping.

Parameters

<i>startEnd</i> The values to set as an Array[2].

Definition at line 232 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.startEnd](#).

```
{this.startEnd = startEnd;}
```

3.2.3.16 void WT.Kindle.Clippings.Clipping.setType (String type)

Sets the type of the clipping.

Parameters

<i>type</i> The value to set.

Definition at line 237 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.type](#).

```
{this.type = type;}
```

3.2.3.17 void WT.Kindle.Clippings.Clipping.setDate (Calendar date)

Sets the date of the clipping.

Parameters

<i>date</i> The value to set.

Definition at line 242 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.date](#).

```
{this.date = date;}
```

3.2.3.18 void WT.Kindle.Clippings.Clipping.setText (String text)

Sets the text of the clipping.

Parameters

<i>text</i> The value to set.

Definition at line 247 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.text](#).

```
{this.text = text;}
```

3.2.3.19 String WT.Kindle.Clippings.Clipping.toXML (int indentation)

Outputs the current clipping in XML format.

Parameters

<i>indentation</i> The number of tabs to indent by.

Returns

A string containing the XML code.

Definition at line 254 of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.author](#), [WT.Kindle.Clippings.Utility.calendarToDateString\(\)](#), [WT.Kindle.Clippings.Utility.calendarToTimeString\(\)](#), [WT.Kindle.Clippings.Clipping.date](#), [WT.Kindle.Clippings.Clipping.startEnd](#), [WT.Kindle.Clippings.Clipping.text](#), and [WT.Kindle.Clippings.Clipping.type](#).

```
        {  
        String indent = "";  
        for(int i = 0;i<indentation;i++){indent += "\t";}  
        String xml = "";
```

```

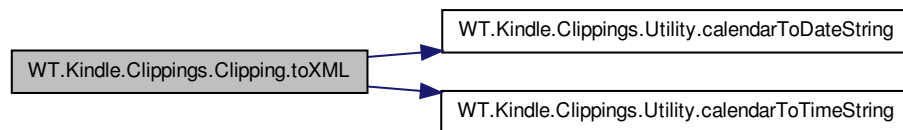
        xml += "\n"      + indent + "<" + type + ">";

        xml += "\n\t"   + indent + "<Author>"   + author
"</Author>";
        xml += "\n\t"   + indent + "<Start>"     + startEnd[0]
"</Start>";
        xml += "\n\t"   + indent + "<End>"       + startEnd[1]
"</End>";
        xml += "\n\t"   + indent + "<Date>"      + Utility.calendarToDateString(
date) + "</Date>";
        xml += "\n\t"   + indent + "<Time>"     + Utility.calendarToTimeString(
date) + "</Time>";
        xml += "\n\t"   + indent + "<Text>"     + text
"</Text>";

        xml += "\n"      + indent + "</" + type + ">";
        return xml;
    }

```

Here is the call graph for this function:



3.2.3.20 String WT.Kindle.Clippings.Clipping.toString ()

Gives a string representation of the [Clipping](#).

Returns

A string containing all the data held in the class.

Definition at line [277](#) of file [Clipping.java](#).

References [WT.Kindle.Clippings.Clipping.author](#), [WT.Kindle.Clippings.Clipping.date](#), [WT.Kindle.Clippings.Clipping.startEnd](#), [WT.Kindle.Clippings.Clipping.text](#), [WT.Kindle.Clippings.Clipping.title](#), and [WT.Kindle.Clippings.Clipping.type](#).

```

        {
        String retval;

        retval = "Clipping[" + title          + ", "

```

```

        + author          + ", "
        + startEnd[0]     + ", "
        + startEnd[1]     + ", "
        + type            + ", "
        + date.toString() + ", "
        + text
    + "];
    return retval;
}

```

3.2.4 Member Data Documentation

3.2.4.1 String WT.Kindle.Clippings.Clipping.author [private]

String to store the author of the clipping.

Definition at line 33 of file [Clipping.java](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#), [WT.Kindle.Clippings.Clipping.getAuthor\(\)](#), [WT.Kindle.Clippings.Clipping.parseAuthor\(\)](#), [WT.Kindle.Clippings.Clipping.setAuthor\(\)](#), [WT.Kindle.Clippings.Clipping.toString\(\)](#), and [WT.Kindle.Clippings.Clipping.toXML\(\)](#).

3.2.4.2 String WT.Kindle.Clippings.Clipping.title [private]

String to store the title of the clipping.

Definition at line 35 of file [Clipping.java](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#), [WT.Kindle.Clippings.Clipping.getTitle\(\)](#), [WT.Kindle.Clippings.Clipping.parseTitle\(\)](#), [WT.Kindle.Clippings.Clipping.setTitle\(\)](#), and [WT.Kindle.Clippings.Clipping.toString\(\)](#).

3.2.4.3 int [] WT.Kindle.Clippings.Clipping.startEnd [private]

Integer array to hold the start and end location of the clipping.

Definition at line 37 of file [Clipping.java](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#), [WT.Kindle.Clippings.Clipping.getStartEnd\(\)](#), [WT.Kindle.Clippings.Clipping.parseStartEnd\(\)](#), [WT.Kindle.Clippings.Clipping.setStartEnd\(\)](#), [WT.Kindle.Clippings.Clipping.toString\(\)](#), and [WT.Kindle.Clippings.Clipping.toXML\(\)](#).

3.2.4.4 String WT.Kindle.Clippings.Clipping.type [private]

String to hold the type of the clipping.

Definition at line 39 of file [Clipping.java](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#), [WT.Kindle.Clippings.Clipping.getType\(\)](#), [WT.Kindle.Clippings.Clipping.parseType\(\)](#), [WT.Kindle.Clippings.Clipping.setType\(\)](#), [WT.Kindle.Clippings.Clipping.toString\(\)](#), and [WT.Kindle.Clippings.Clipping.toXML\(\)](#).

3.2.4.5 Calendar WT.Kindle.Clippings.Clipping.date [private]

Calendar to store the date and time of the clipping.

Definition at line 41 of file [Clipping.java](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#), [WT.Kindle.Clippings.Clipping.getDate\(\)](#), [WT.Kindle.Clippings.Clipping.parseDateTime\(\)](#), [WT.Kindle.Clippings.Clipping.setDate\(\)](#), [WT.Kindle.Clippings.Clipping.toString\(\)](#), and [WT.Kindle.Clippings.Clipping.toXML\(\)](#).

3.2.4.6 String WT.Kindle.Clippings.Clipping.text [private]

String to hold the text of the clipping. Either the highlighted text or the user's note.

Definition at line 43 of file [Clipping.java](#).

Referenced by [WT.Kindle.Clippings.Clipping.Clipping\(\)](#), [WT.Kindle.Clippings.Clipping.getText\(\)](#), [WT.Kindle.Clippings.Clipping.parseText\(\)](#), [WT.Kindle.Clippings.Clipping.setText\(\)](#), [WT.Kindle.Clippings.Clipping.toXML\(\)](#), and [WT.Kindle.Clippings.Clipping.toXML\(\)](#).

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

- [Clipping.java](#)

3.3 WT.Kindle.Clippings.Utility Class Reference

The class contains useful methods for the ClippingsParser library. Methods are provided to parse the "My Clippings.txt" file found on kindle devices into a useful format using the [Clipping](#) and [Book](#) classes.

Static Public Member Functions

- static String [calendarToDateString](#) (Calendar cal)
Converts a Calendar object into a date formatted string.
- static String [calendarToTimeString](#) (Calendar cal)
Converts a Calendar object into a time formatted string.
- static [Clipping\[\]](#) [parseClippings](#) (String input)
Parses a number of clippings contained in one string into an Array of Clipping.

- static `ArrayList< Book > generateBooksList` (String input)
Generates an ArrayList of Book from a string containing clippings.
- static `String generateBooksXML` (String input)
Generates an XML representation of the clippings.
- static `String generateBooksXML` (ArrayList< Book > input)
Generates an XML representation of the clippings.
- static `void debug` (String classs, String location, String variableName, String variable)
A method to output debug information in a uniform way.
- static `void debug` (String classs, String location, String message)
A method to output debug information in a uniform way.
- static `boolean isDebugEnabled` ()
A method to check if the debug methods are turned on at runtime.
- static `String fileToString` (String filePath)
Loads a file from disk into a string.

Static Private Attributes

- static `boolean debugON` = false
A boolean value to turn the debug method on and off.

3.3.1 Detailed Description

The class contains useful methods for the ClippingsParser library. Methods are provided to parse the "My Clippings.txt" file found on kindle devices into a useful format using the `Clipping` and `Book` classes.

Date

Created: 26/08/2011
Modified: 27/08/2011

Author

Antony Jordan

Todo

Add code to handle errors such as incorrect file layout.

Definition at line 43 of file `Utility.java`.

3.3.2 Member Function Documentation

3.3.2.1 static String WT.Kindle.Clippings.Utility.calendarToDateString (Calendar *cal*) [static]

Converts a Calendar object into a date formatted string.

Parameters

<i>cal</i> The calendar to convert.

Returns

The String representation of the date.

Definition at line 49 of file [Utility.java](#).

Referenced by [WT.Kindle.Clippings.Clipping.toXML\(\)](#).

```
int intDay =cal.get(Calendar.DAY_OF_MONTH);
int intMonth =cal.get(Calendar.MONTH);
int intYear = cal.get(Calendar.YEAR);

intMonth++;

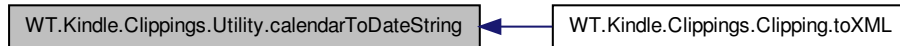
String strDay;
String strMonth;

if(intDay < 10){
    strDay = "0" + intDay;
}
else{
    strDay = "" + intDay;
}
if(intMonth < 10){
    strMonth = "0" + intMonth;
}
else{
    strMonth = "" + intMonth;
}

String retVal = strDay + "/" + strMonth + "/" + intYear;

return retVal;
}
```

Here is the caller graph for this function:



3.3.2.2 static String WT.Kindle.Clippings.Utility.calendarToTimeString (Calendar *cal*) [static]

Converts a Calendar object into a time formatted string.

Parameters

cal The calendar to convert.

Returns

The String representation of the time.

Definition at line 81 of file [Utility.java](#).

Referenced by [WT.Kindle.Clippings.Clipping.toXML\(\)](#).

```
int intHour = cal.get(Calendar.HOUR_OF_DAY);
int intMin = cal.get(Calendar.MINUTE);

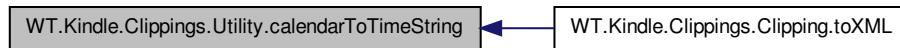
String strHour;
String strMin;

if(intHour < 10){
    strHour = "0" + intHour;
}
else{
    strHour = "" + intHour;
}
if(intMin < 10){
    strMin = "0" + intMin;
}
else{
    strMin = "" + intMin;
}

String retVal = strHour + ":" + strMin;

return retVal;
}
```


Here is the caller graph for this function:



3.3.2.3 static Clipping [] WT.Kindle.Clippings.Utility.parseClippings (String input) [static]

Parses a number of clippings contained in one string into an Array of Clipping.

Parameters

<i>input</i>	The string containing the multiple clippings.
--------------	---

Returns

An Array of [Clipping](#) containing all of the clippings.

Definition at line 111 of file [Utility.java](#).

Referenced by [WT.Kindle.Clippings.Utility.generateBooksList\(\)](#).

```

    {
        ArrayList<Clipping> retVal = new ArrayList<Clipping>();

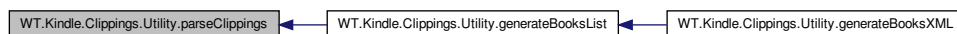
        String delimiter = "=====";

        input = input.substring(0, input.lastIndexOf(delimiter));
        String[] clippings = input.split(delimiter);

        for (String clipping : clippings) {
            Clipping clip = new Clipping(clipping.trim());
            retVal.add(clip);
        }

        return retVal.toArray(new Clipping[]{});
    }
  
```

Here is the caller graph for this function:



3.3.2.4 static ArrayList<Book> WT.Kindle.Clippings.Utility.generateBooksList (String input) [static]

Generates an ArrayList of [Book](#) from a string containing clippings.

Parameters

<i>input</i>	The string containing the clippings.
--------------	--------------------------------------

Returns

An ArrayList of [Book](#) containign the clippings.

Definition at line 131 of file [Utility.java](#).

References [WT.Kindle.Clippings.Book.getClippings\(\)](#), [WT.Kindle.Clippings.Clipping.getTitle\(\)](#), and [WT.Kindle.Clippings.Utility.parseClippings\(\)](#).

Referenced by [WT.Kindle.Clippings.Utility.generateBooksXML\(\)](#).

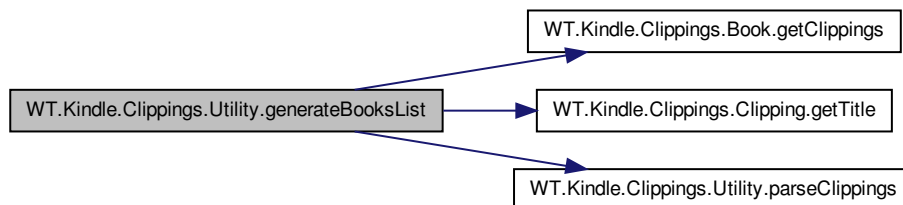
```
Clipping[] clippings = Utility.parseClippings(input);
ArrayList<Book> books = new ArrayList<Book>();

for (Clipping clipping : clippings) {
    String clippingTitle = clipping.getTitle();

    boolean added = false;

    for (Book book : books) {
        if (clippingTitle.equals(book.getTitle()))
        {
            book.getClippings().add(clipping);
            added = true;
        }
    }
    if (!added) {
        Book book = new Book(clippingTitle);
        book.getClippings().add(clipping);
        books.add(book);
    }
}
return books;
}
```

Here is the call graph for this function:



Here is the caller graph for this function:



3.3.2.5 static String WT.Kindle.Clippings.Utility.generateBooksXML (String input) [static]

Generates an XML representation of the clippings.

Parameters

<i>input</i>	The string containing the clippings.
--------------	--------------------------------------

Returns

A string of XML code containing the clippings

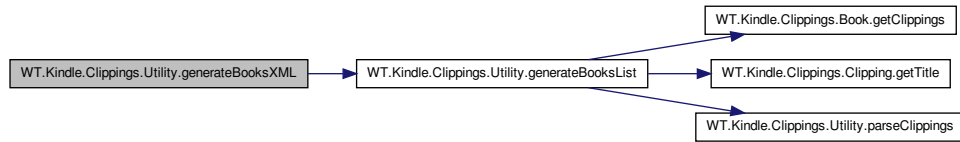
Definition at line 161 of file [Utility.java](#).

References [WT.Kindle.Clippings.Utility.generateBooksList\(\)](#).

```

    String xml = generateBooksXML(generateBooksList(input));
    return xml;
}
  
```

Here is the call graph for this function:



3.3.2.6 static String WT.Kindle.Clippings.Utility.generateBooksXML (ArrayList< Book > *input*) [static]

Generates an XML representation of the clippings.

Parameters

input An ArrayList of [Book](#).

Returns

A string of XML code containing the clippings

Definition at line 171 of file [Utility.java](#).

```

    {
        ArrayList<Book> books = input;

        String xml;

        xml = "<Clippings>";

        for (Book book : books) {
            xml += book.toXML(1);
        }

        xml += "\n";
        xml += "</Clippings>";

        return xml;
    }
  
```

3.3.2.7 static void WT.Kindle.Clippings.Utility.debug (String *class*, String *location*, String *variableName*, String *variable*) [static]

A method to output debug information in a uniform way.

Parameters

<i>class</i>	The class name that the message originates in.
<i>location</i>	The location within the class e.g. method name.
<i>variable-Name</i>	The name of the variable to display.
<i>variable</i>	The actual variable to show.

Definition at line 197 of file [Utility.java](#).

References [WT.Kindle.Clippings.Utility.isDebugEnabled\(\)](#).

Referenced by [WT.Kindle.Clippings.Utility.fileToString\(\)](#), [WT.Kindle.Clippings.Clipping.parseAuthor\(\)](#), and [WT.Kindle.Clippings.Clipping.parseDateTime\(\)](#).

```

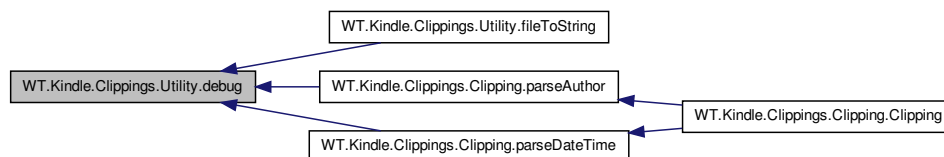
    {
        if(isDebugEnabled()) {System.out.println("Debug (" +class+" "+location+" ): "+variableName+" = "+variable);}
    }

```

Here is the call graph for this function:



Here is the caller graph for this function:



3.3.2.8 static void WT.Kindle.Clippings.Utility.debug (String *class*, String *location*, String *message*) [static]

A method to output debug information in a uniform way.

Parameters

<i>class</i>	The class name that the message originates in.
<i>location</i>	The location within the class e.g. method name.
<i>message</i>	The message to display.

Definition at line 206 of file [Utility.java](#).

References [WT.Kindle.Clippings.Utility.isDebugEnabled\(\)](#).

```

        if(isDebugEnabled()) {System.out.println("Debug (" +class+" ":"+location+" ): "+mes
        sage); }
    }

```

Here is the call graph for this function:

**3.3.2.9 static boolean WT.Kindle.Clippings.Utility.isDebugEnabled () [static]**

A method to check if the debug methods are turned on at runtime.

Returns

True for yes, false for no.

Definition at line 213 of file [Utility.java](#).

References [WT.Kindle.Clippings.Utility.debugON](#).

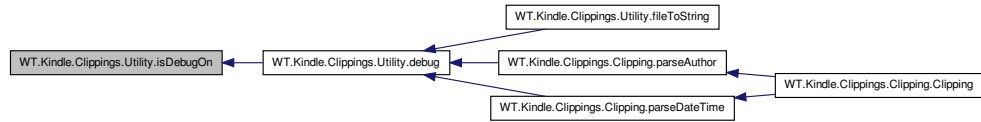
Referenced by [WT.Kindle.Clippings.Utility.debug\(\)](#).

```

{ return debugON; }

```

Here is the caller graph for this function:



3.3.2.10 static String WT.Kindle.Clippings.Utility.fileToString (String filePath) [static]

Loads a file from disk into a string.

Parameters

<i>filePath</i> The path to the file.

Returns

The string holding the files contents.

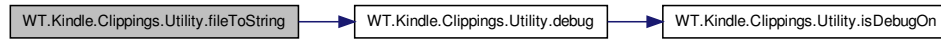
Definition at line 220 of file [Utility.java](#).

References [WT.Kindle.Clippings.Utility.debug\(\)](#).

```

    {
    byte[] buffer = new byte[(int) new File(filePath).length()];
    FileInputStream f;
    try {
        f = new FileInputStream(filePath);
        f.read(buffer);
        f.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    Utility.debug("Utility", "fileToString", "buffer", new String(buf
fer));
    return new String(buffer);
    }
  
```

Here is the call graph for this function:



3.3.3 Member Data Documentation

3.3.3.1 boolean `WT.Kindle.Clippings.Utility.debugON = false` [`static`, `private`]

A boolean value to turn the debug method on and off.

Definition at line 189 of file [Utility.java](#).

Referenced by [WT.Kindle.Clippings.Utility.isDebugEnabled\(\)](#).

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

- [Utility.java](#)

4 File Documentation

4.1 Book.java File Reference

This file holds the source code for the Book class.

Classes

- class [WT.Kindle.Clippings.Book](#)

The [Book](#) class stores all the clippings from a particular book title in a single location. It uses the [Clipping](#) class to store the clippings as an object in an `ArrayList` and stores their common title as a single string.

Packages

- package [WT.Kindle.Clippings](#)

Package containing the code for the [ClippingsParser](#) library.

4.1.1 Detailed Description

This file holds the source code for the Book class.

Date

Created: 25/08/2011
Modified: 27/08/2011

Author

Antony Jordan

Definition in file [Book.java](#).

4.2 Book.java

```
00001 /*
00002  * This work is licensed under the Creative Commons Attribution-NonCommercial-Sha
00003  * reAlike 3.0 Unported License.
00004  * To view a copy of this license, visit http://creativecommons.org/licenses/by-n
00005  * c-sa/3.0/ or send a letter
00006  * to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California,
00007  * 94041, USA.
00008  */
00009 package WT.Kindle.Clippings;
00010 import java.util.ArrayList;
00011 public class Book {
00012     private String title;
00013     private ArrayList<Clipping> clippings;
00014     private boolean isTitleSet;
00015     public Book() {
00016         this.title = "";
00017         clippings = new ArrayList<Clipping>();
00018         isTitleSet = false;
00019     }
00020     public Book(String title) {
00021         this.title = title;
00022         clippings = new ArrayList<Clipping>();
00023         isTitleSet = true;
00024     }
00025     public boolean IsTitleSet(){return isTitleSet;}
00026     public void add(Clipping clipping){
00027         if(title.equals(clipping.getTitle())){
00028             clippings.add(clipping);
00029         }
00030         else{
00031         }
00032     }
00033     public void setTitle(String title){
00034         if(!isTitleSet){
00035             this.title = title;
00036         }
00037     }
00038 }
```

```

00077         }
00078         else{
00080         }
00081     }
00086     public void setClippings (ArrayList<Clipping> clippings) {this.clippings =
clippings;}
00087
00092     public String getTitle() {return title;}
00097     public ArrayList<Clipping> getClippings() {return clippings;}
00098
00104     public String toXML(int indentation){
00105         String indent = "";
00106         for(int i = 0;i<indentation;i++){indent += "\t";}
00107
00108         String xml = "";
00109
00110         xml += "\n"      + indent + "<Book>";
00111
00112         xml += "\n\t"   + indent + "<Title>" + title + "</Title>";
00113
00114         for (Clipping clipping : clippings) {
00115             xml+=clipping.toXML(indentation + 1);
00116         }
00117
00118         xml += "\n"      + indent + "</Book>";
00119
00120         return xml;
00121     }
00123     public String toString() {
00124         String retVal;
00125
00126         retVal = title + "(" + clippings.size() + ")";
00127
00128         return retVal;
00129     }
00130 }

```

4.3 Clipping.java File Reference

This file holds the source code for the Clipping class.

Classes

- class [WT.Kindle.Clippings.Clipping](#)

*The class stores the data for a single clipping from the "My Clippings.txt" of the kin-
dle. Methods are provided to parse each item in the clipping into individual objects.*

Packages

- package [WT.Kindle.Clippings](#)

Package containing the code for the ClippingsParser library.

4.3.1 Detailed Description

This file holds the source code for the Clipping class.

Date

Created: 25/08/2011
Modified: 27/08/2011

Author

Antony Jordan

Definition in file [Clipping.java](#).

4.4 Clipping.java

```
00001 /*
00002  * This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.
00003  * To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter
00004  * to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.
00005  */
00006
00007 package WT.Kindle.Clippings;
00008
00009 import java.text.DateFormat;
00010 import java.text.ParseException;
00011 import java.text.SimpleDateFormat;
00012 import java.util.Calendar;
00013
00030 public class Clipping
00031 {
00033     private String author;
00035     private String title;
00037     private int[] startEnd;
00039     private String type;
00041     private Calendar date;
00043     private String text;
00044
00046     public Clipping() {
00047         author = null;
00048         title = null;
00049         startEnd = null;
00050         type = null;
00051         date = Calendar.getInstance();
00052         text = null;
00053     }
00058     public Clipping(String input) {
00059         author = parseAuthor(input);
00060         title = parseTitle(input);
00061         startEnd = parseStartEnd(input);
00062         type = parseType(input);
00063         date = Calendar.getInstance();
00064         date = parseDateTime(input);
00065         text = parseText(input);
```

```
00066     }
00067
00073     private String parseAuthor(String input) {
00074         String temp = input.substring(0, input.indexOf("\n"));
00075
00076         Utility.debug("Clipping", "parseAuthor", "temp",temp);
00077
00078         Integer authorStartIndex = temp.lastIndexOf("(");
00079
00080         Utility.debug("Clipping", "parseAuthor", "authorStartIndex",authorStartIn
dex.toString());
00081
00082         authorStartIndex++;
00083
00084         Utility.debug("Clipping", "parseAuthor", "authorStartIndex",authorStartIn
dex.toString());
00085
00086         Integer authorEndIndex = (temp.lastIndexOf(")"));
00087
00088         Utility.debug("Clipping", "parseAuthor", "authorEndIndex",authorEndIndex.
toString());
00089
00090         String author = temp.substring(authorStartIndex, authorEndIndex);
00091
00092         author = author.trim();
00093
00094         return author;
00095     }
00101     private String parseTitle(String input) {
00102         String temp = input.substring(0, input.indexOf("\n"));
00103         int authorStartIndex = temp.lastIndexOf("(");
00104
00105         String title = temp.substring(0, authorStartIndex);
00106
00107         title = title.trim();
00108
00109         return title;
00110     }
00116     private int[] parseStartEnd(String input) {
00117         int[] retVal = new int[2];
00118         String startEnd = input.substring(input.indexOf("Loc.") + 5,input.indexOf
("|"));
00119
00120         startEnd = startEnd.trim();
00121         if (startEnd.contains("-")) {
00122             String[] temp = startEnd.split("-");
00123
00124             temp[0] = temp[0].trim();
00125             temp[1] = temp[1].trim();
00126
00127             retVal[0] = Integer.parseInt(temp[0]);
00128             retVal[1] = Integer.parseInt(temp[1]);
00129         }
00129         else
00130         {
00131             retVal[0] = retVal[1] = Integer.parseInt(startEnd);
00132         }
00133         return retVal;
00134     }
00140     private String parseType(String input) {
00141         String type = input.substring(input.indexOf("\n") + 3,input.indexOf("Loc.
"));
00142     }
```

```

00143         type = type.trim();
00144
00145         return type;
00146     }
00152     private Calendar parseDateTime(String input) {
00153         String date = input.substring(input.indexOf("|"));
00154
00155         Utility.debug("Clipping", "parseDateTime", "date", date.replace("\n", "#"
    ));
00156
00157         if(date.contains("\n")){
00158             date = date.substring(0,date.indexOf("\n"));
00159         }
00160         date = date.replace("| Added on ", "");
00161
00162         DateFormat formatter = new SimpleDateFormat("EEEE, MMMM dd, yyyy, hh:mm a
    ");
00163
00164         Calendar calendar = Calendar.getInstance();
00165         try {
00166             calendar.setTime(formatter.parse(date));
00167         }
00168         catch (ParseException e) {
00169             e.printStackTrace();
00170         }
00171
00172         return calendar;
00173     }
00179     private String parseText (String input) {
00180         String text = "";
00181         if(text.contains("\n")){text = input.substring((input.indexOf("\n", input
    .indexOf("\n") + 1));}
00182         text = text.trim();
00183
00184         return text;
00185     }
00186
00191     public String getAuthor(){return author;}
00196     public String getTitle(){return title;}
0201     public int[] getStartEnd(){return startEnd;}
0206     public String getType(){return type;}
0211     public Calendar getDate(){return date;}
0216     public String getText(){return text;}
0217
0222     public void setAuthor(String author){this.author = author;}
0227     public void setTitle(String title){this.title = title;}
0232     public void setStartEnd(int[] startEnd){this.startEnd = startEnd;}
0237     public void setType(String type){this.type = type;}
0242     public void setDate(Calendar date){this.date = date;}
0247     public void setText(String text){this.text = text;}
0248
0254     public String toXML(int indentation){
0255         String indent = "";
0256         for(int i = 0;i<indentation;i++){indent += "\t";}
0257
0258         String xml = "";
0259
0260         xml += "\n"      + indent + "<" + type + ">";
0261
0262         xml += "\n\t"    + indent + "<Author>"  + author
    "</Author>";
0263         xml += "\n\t"    + indent + "<Start>"    + startEnd[0]

```

```

    "</Start>";
00264     xml += "\n\t" + indent + "<End>" + startEnd[1]
"</End>";
00265     xml += "\n\t" + indent + "<Date>" + Utility.calendarToDateString(
date) + "</Date>";
00266     xml += "\n\t" + indent + "<Time>" + Utility.calendarToTimeString(
date) + "</Time>";
00267     xml += "\n\t" + indent + "<Text>" + text
"</Text>";

00268
00269     xml += "\n" + indent + "</" + type + ">";
00270     return xml;
00271 }
00272
00273 public String toString() {
00274     String retval;
00275
00276     retval = "Clipping[" + title + ", "
00277             + author + ", "
00278             + startEnd[0] + ", "
00279             + startEnd[1] + ", "
00280             + type + ", "
00281             + date.toString() + ", "
00282             + text
00283             + " ]";
00284
00285     return retval;
00286 }
00287 }
00288 }
00289 }
00290 }
00291 }

```

4.5 Utility.java File Reference

This file holds the source code for the Utility class.

Classes

- class [WT.Kindle.Clippings.Utility](#)

The class contains useful methods for the ClippingsParser library. Methods are provided to parse the "My Clippings.txt" file found on kindle devices into a useful format using the [Clipping](#) and [Book](#) classes.

Packages

- package [WT](#)

Package containing code from developer Antony Jordan.

- package [WT.Kindle](#)

Package containing all code relating to the kindle.

- package [WT.Kindle.Clippings](#)

Package containing the code for the ClippingsParser library.

4.5.1 Detailed Description

This file holds the source code for the Utility class.

Date

Created: 26/08/2011
Modified: 27/08/2011

Author

Antony Jordan

Definition in file [Utility.java](#).

4.6 Utility.java

```
00001 /*
00002  * This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.
00003  * To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter
00004  * to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.
00005  */
00006
00019 package WT.Kindle.Clippings;
00020
00021 import java.io.File;
00022 import java.io.FileInputStream;
00023 import java.io.IOException;
00024 import java.util.ArrayList;
00025 import java.util.Calendar;
00026
00043 public class Utility {
00049     public static String calendarToDateString(Calendar cal){
00050         int intDay =cal.get(Calendar.DAY_OF_MONTH);
00051         int intMonth =cal.get(Calendar.MONTH);
00052         int intYear = cal.get(Calendar.YEAR);
00053
00054         intMonth++;
00055
00056         String strDay;
00057         String strMonth;
00058
00059         if(intDay < 10){
00060             strDay = "0" + intDay;
00061         }
00062         else{
00063             strDay = "" + intDay;
00064         }
00065         if(intMonth < 10){
00066             strMonth = "0" + intMonth;
00067         }
00068         else{
00069             strMonth = "" + intMonth;
00070         }
00071     }
```

```
00072     String retVal = strDay + "/" + strMonth + "/" + intYear;
00073
00074     return retVal;
00075 }
00081 public static String calendarToTimeString(Calendar cal) {
00082     int intHour = cal.get(Calendar.HOUR_OF_DAY);
00083     int intMin = cal.get(Calendar.MINUTE);
00084
00085     String strHour;
00086     String strMin;
00087
00088     if(intHour < 10){
00089         strHour = "0" + intHour;
00090     }
00091     else{
00092         strHour = "" + intHour;
00093     }
00094     if(intMin < 10){
00095         strMin = "0" + intMin;
00096     }
00097     else{
00098         strMin = "" + intMin;
00099     }
00100
00101     String retVal = strHour + ":" + strMin;
00102
00103     return retVal;
00104 }
00105
00111 public static Clipping[] parseClippings(String input) {
00112     ArrayList<Clipping> retVal = new ArrayList<Clipping>();
00113
00114     String delimiter = "=====";
00115
00116     input = input.substring(0,input.lastIndexOf(delimiter));
00117     String[] clippings = input.split(delimiter);
00118
00119     for (String clipping : clippings) {
00120         Clipping clip = new Clipping(clipping.trim());
00121         retVal.add(clip);
00122     }
00123
00124     return retVal.toArray(new Clipping[]{});
00125 }
00131 public static ArrayList<Book> generateBooksList(String input) {
00132     Clipping[] clippings = Utility.parseClippings(input);
00133     ArrayList<Book> books = new ArrayList<Book>();
00134
00135     for (Clipping clipping : clippings) {
00136         String clippingTitle = clipping.getTitle();
00137
00138         boolean added = false;
00139
00140         for (Book book : books) {
00141             if (clippingTitle.equals(book.getTitle()))
00142             {
00143                 book.getClippings().add(clipping);
00144                 added = true;
00145             }
00146         }
00147         if (!added) {
00148             Book book = new Book(clippingTitle);
```



```
00149         book.getClippings().add(clipping);
00150         books.add(book);
00151     }
00152 }
00153     return books;
00154 }
00155
00161     public static String generateBooksXML(String input) {
00162         String xml = generateBooksXML(generateBooksList(input));
00163
00164         return xml;
00165     }
00171     public static String generateBooksXML(ArrayList<Book> input) {
00172         ArrayList<Book> books = input;
00173
00174         String xml;
00175
00176         xml = "<Clippings>";
00177
00178         for (Book book : books) {
00179             xml += book.toXML(1);
00180         }
00181
00182         xml += "\n";
00183         xml += "</Clippings>";
00184
00185         return xml;
00186     }
00187
00189     private static boolean debugON = false;
00197     public static void debug(String classes,String location,String variableName,
00198     String variable){
00199         if(isDebugEnabled()){System.out.println("Debug("+classes+": "+location+"): "+variableName+" = "+variable);}
00199     }
00206     public static void debug(String classes,String location,String message){
00207         if(isDebugEnabled()){System.out.println("Debug("+classes+": "+location+"): "+message);}
00208     }
00213     public static boolean isDebugEnabled(){return debugON;}
00214
00220     public static String fileToString(String filePath){
00221         byte[] buffer = new byte[(int) new File(filePath).length()];
00222
00223         FileInputStream f;
00224
00225         try {
00226             f = new FileInputStream(filePath);
00227
00228             f.read(buffer);
00229             f.close();
00230         } catch (IOException e) {
00231             e.printStackTrace();
00232         }
00233
00234         Utility.debug("Utility", "fileToString", "buffer", new String(buffer));
00235
00236         return new String(buffer);
00237     }
00238 }
```

Index

- add
 - WT::Kindle::Clippings::Book, 5
- author
 - WT::Kindle::Clippings::Clipping, 26
- Book
 - WT::Kindle::Clippings::Book, 4, 5
- Book.java, 38
- calendarToDateString
 - WT::Kindle::Clippings::Utility, 29
- calendarToTimeString
 - WT::Kindle::Clippings::Utility, 30
- Clipping
 - WT::Kindle::Clippings::Clipping, 13
- Clipping.java, 40
- clippings
 - WT::Kindle::Clippings::Book, 9
- date
 - WT::Kindle::Clippings::Clipping, 27
- debug
 - WT::Kindle::Clippings::Utility, 34, 35
- debugON
 - WT::Kindle::Clippings::Utility, 38
- fileToString
 - WT::Kindle::Clippings::Utility, 37
- generateBooksList
 - WT::Kindle::Clippings::Utility, 31
- generateBooksXML
 - WT::Kindle::Clippings::Utility, 33, 34
- getAuthor
 - WT::Kindle::Clippings::Clipping, 20
- getClippings
 - WT::Kindle::Clippings::Book, 7
- getDate
 - WT::Kindle::Clippings::Clipping, 21
- getStartEnd
 - WT::Kindle::Clippings::Clipping, 21
- getText
 - WT::Kindle::Clippings::Clipping, 22
- getTitle
 - WT::Kindle::Clippings::Book, 7
 - WT::Kindle::Clippings::Clipping, 20
- getType
 - WT::Kindle::Clippings::Clipping, 21
- isDebugEnabled
 - WT::Kindle::Clippings::Utility, 36
- IsTitleSet
 - WT::Kindle::Clippings::Book, 5
- isTitleSet
 - WT::Kindle::Clippings::Book, 9
- parseAuthor
 - WT::Kindle::Clippings::Clipping, 14
- parseClippings
 - WT::Kindle::Clippings::Utility, 31
- parseDateTime
 - WT::Kindle::Clippings::Clipping, 18
- parseStartEnd
 - WT::Kindle::Clippings::Clipping, 16
- parseText
 - WT::Kindle::Clippings::Clipping, 19
- parseTitle
 - WT::Kindle::Clippings::Clipping, 15
- parseType
 - WT::Kindle::Clippings::Clipping, 17
- setAuthor
 - WT::Kindle::Clippings::Clipping, 22
- setClippings
 - WT::Kindle::Clippings::Book, 7
- setDate
 - WT::Kindle::Clippings::Clipping, 23
- setStartEnd
 - WT::Kindle::Clippings::Clipping, 23
- setText
 - WT::Kindle::Clippings::Clipping, 24
- setTitle
 - WT::Kindle::Clippings::Book, 6
 - WT::Kindle::Clippings::Clipping, 22
- setType
 - WT::Kindle::Clippings::Clipping, 23
- startEnd
 - WT::Kindle::Clippings::Clipping, 26
- text
 - WT::Kindle::Clippings::Clipping, 27
- title
 - WT::Kindle::Clippings::Book, 9
 - WT::Kindle::Clippings::Clipping, 26
- toString
 - WT::Kindle::Clippings::Book, 9

- WT::Kindle::Clippings::Clipping, 25
- toXML
 - WT::Kindle::Clippings::Book, 8
 - WT::Kindle::Clippings::Clipping, 24
- type
 - WT::Kindle::Clippings::Clipping, 26
- Utility.java, 44
- WT, 1
- WT.Kindle, 1
- WT.Kindle.Clippings, 2
- WT::Kindle::Clippings::Book, 2
 - add, 5
 - Book, 4, 5
 - clippings, 9
 - getClippings, 7
 - getTitle, 7
 - IsTitleSet, 5
 - isTitleSet, 9
 - setClippings, 7
 - setTitle, 6
 - title, 9
 - toString, 9
 - toXML, 8
- WT::Kindle::Clippings::Clipping, 10
 - author, 26
 - Clipping, 13
 - date, 27
 - getAuthor, 20
 - getDate, 21
 - getStartEnd, 21
 - getText, 22
 - getTitle, 20
 - getType, 21
 - parseAuthor, 14
 - parseDateTime, 18
 - parseStartEnd, 16
 - parseText, 19
 - parseTitle, 15
 - parseType, 17
 - setAuthor, 22
 - setDate, 23
 - setStartEnd, 23
 - setText, 24
 - setTitle, 22
 - setType, 23
 - startEnd, 26
 - text, 27
 - title, 26
 - toString, 25
 - toXML, 24
 - type, 26
- WT::Kindle::Clippings::Utility, 27
 - calendarToDateString, 29
 - calendarToTimeString, 30
 - debug, 34, 35
 - debugON, 38
 - fileToString, 37
 - generateBooksList, 31
 - generateBooksXML, 33, 34
 - isDebugOn, 36
 - parseClippings, 31