IniFile

This object allows you to read and write to a INI file.

Quick Use

Usable example:

```
var myObject= new IniFile();
myObject.load("config.ini");
myObject.setSection("SomeSettingsSection");
if (myObject.keyExists("SomeKey")){
    mytext = myObject.keyValue("SomeKey");
    newtext = mytext + "Bar";
    myObject.setKeyValue("SomeKey", newtext);
}else{
    myObject.setKeyValue("SomeKey", "Foo" );
    mytext = cfgini.keyValue("SomeKey");
}
myObject.save("config.ini");
```

Will result in value "Bar" being appended ever run, except first. Defaults to "Foo" if Key is unset:

Functions

IniFile

Constructor function.

Syntax

```
var myObject = new IniFile();
```

```
var myObject = new IniFile(parameters);
```

Arguments

1. parameters - (object) parameters
    - delimiter - (char) the delimiter to use
    - commentchar - (char) the comment character
    - encoding - (Encoding) the encoding to use

Exceptions

- (ParameterTypeError) incorrect parameter type
- (ParameterCountError) incorrect parameter count

Example

Create a IniFile object.

```
var myObject = new IniFile();
```

Create a IniFile object with parameters.

```
var myObject = new IniFile({
    delimiter: "=",
    commentCharacter: "$"
});
```

Methods

load

Loads a file.

Syntax

```
myObject.load(filename);
```

Arguments

1. filename - (string) the filename of the file to open

Returns

- (IniFile) this IniFile

Exceptions

- (LoadFileError) cannot load the file

save

Saves a file.

Syntax

```
myObject.save(filename);
```

Arguments

1. filename - (string) the filename of the file to save

Returns

- (IniFile) this IniFile

Exceptions

- (SaveFileError) cannot save the file

clear

Clear the file.

Syntax

```
myObject.clear();
```

Returns

- (IniFile) this IniFile

preserveDeletedData

Preserve the deleted data as comments.

Syntax

```
myObject.preserveDeletedData(preserve);
```

Arguments

1. preserve - (boolean) should the deleted data preserved as comments

Returns

- (IniFile) this IniFile

setDelimiter

Set the value/data delimiter. Default is =.

Syntax

```
myObject.setDelimiter(delimiter);
```

Arguments

1. delimiter - (char) the delimiter to use

Returns

- (IniFile) this IniFile

setCommentCharacter

Set the comment character. Default is #.

Syntax

```
myObject.setCommentCharacter(commentchar);
```

Arguments

1. commentchar - (char) the comment character

Returns

- (IniFile) this IniFile

setSection

Sets the current INI section.

Syntax

```
myObject.setSection(sectionName);
```

```
myObject.setSection(sectionName, create);
```

Arguments

1. sectionName - (string) the section name
2. create - (boolean) create the section if it doesn't exists (default: true)

Returns

- (IniFile) this IniFile

Exceptions

- (FindSectionError) cannot find the section

setEncoding

Sets the encoding to use.

Syntax

```
myObject.setEncoding(encoding);
```

Arguments

1. encoding - (Encoding) the encoding to use

Returns

- (IniFile) this IniFile

sectionAt

Returns the section name at index **sectionIndex**.

Syntax

```
myObject.sectionAt(sectionIndex);
```

Arguments

1. sectionIndex - (integer) the section index

Returns

- (string) the section name

Exceptions

- (FindSectionError) cannot find the section

deleteSection

Deletes a section.

Syntax

```
myObject.deleteSection(sectionName);
```

Arguments

1. sectionName - (string) the section name

Returns

- (IniFile) this IniFile

Exceptions

- (FindSectionError) cannot find the section

sectionCount

Returns the number of sections.

Syntax

```
myObject.sectionCount();
```

Returns

- (integer) the section count

keyExists

Returns true if a key with **keyName** exists.

Syntax

```
myObject.keyExists(keyName);
```

Arguments

1. keyName - (string) the key name

Returns

- (boolean) true if a key with **keyName** exists

keyAt

Returns the name of the key at index **keyIndex**.

Syntax

```
myObject.keyAt(keyIndex);
```

Arguments

1. keyIndex - (integer) the key index

Returns

- (string) the key name

Exceptions

- (KeyError) cannot find the key

keyValue

Returns the value of the key **keyName**.

Syntax

```
myObject.keyValue(keyName);
```

Arguments

1. keyName - (string) the key name

Returns

- (string) the key value

Exceptions

- (KeyError) cannot find the key

setKeyValue

Sets the value of the key **keyName**.

Syntax

```
myObject.setKeyValue(keyName, value);
```

Arguments

1. keyName - (string) the key name
2. value - (string) the key value

Returns

- (IniFile) this IniFile

deleteKey

Delete the key **keyName**.

Syntax

```
myObject.deleteKey(keyName);
```

Arguments

1. keyName - (string) the key name

Returns

- (IniFile) this IniFile

Exceptions

- (KeyError) cannot find the key

keyCount

Returns the key count.

Syntax

```
myObject.keyCount();
```

Returns

- (integer) the key count