

# Printy GUI Namespace

# Table of Contents

## Symbol Reference

### net.stefanhauser.Printy.GUI Namespace

#### AbstractPrinterLogic Class

##### Fields

formattedOutput Field

parameters Field

tmpInFile Field

TmpInFile Field

tmpOutFile Field

TmpOutFile Field

##### Constructors

AbstractPrinterLogic Constructor

##### Methods

CreateFormattedOutput Method

DeleteTempFiles Method

FormatOutput Method

OnInputFileChanged Method

OnOutputFileChanged Method

ReadFromInputFile Method

ReadOutputFromFile Method

WriteToOutputFile

##### Properties

CommandLine Property

FormattedOutput Property

InputFile Property

Location Property

OutputFile Property

ParameterLine Property

##### Events

OutputFileChanged Event

#### Buffer Class

##### Fields

EOF Field

##### Constructors

Buffer Constructor

##### Methods

1

1

2

3

3

3

3

4

4

4

4

4

4

4

4

4

5

5

5

5

5

5

6

6

6

6

6

6

6

6

6

7

7

7

7

8

GetString Method	8
Peek Method	8
Read Method	8
Properties	8
Pos Property	8
CustomPrinterLogic Class	8
Methods	10
FormatOutput Method	10
Properties	10
InputFile Property	10
DisplayGrammarControl Class	10
Fields	11
console Field	11
Gap Field	11
refresh Field	11
Rtb Field	12
Constructors	12
DisplayGrammarControl Constructor	12
Methods	12
Dispose Method	12
OnSizeChanged Method	12
Errors Class	12
Fields	13
count Field	13
errMsgFormat Field	13
Methods	13
Error Method	13
Exception Method	13
SemErr Method	13
SynErr Method	14
ExecutablePrinterLogic Class	14
Methods	15
FormatOutput Method	15
Properties	15
ApplicationName Property	16
HTMLForm Class	16
Constructors	16
HTMLForm Constructor	17
Methods	17
Dispose Method	17

Reset Method	17
SetParameterEventHandler Method	17
SetupHelpProvider Method	17
Properties	17
AdjustedParameter Property	17
Parameter Property	17
LibraryPrinterLogic Class	17
Methods	19
FormatOutput Method	19
MainClass Class	19
Methods	20
Main Method	20
MainForm Class	20
Constructors	20
MainForm	20
Methods	21
InitializeCustomComponents Method	21
OnLoad Method	21
Properties	21
AssemblyLocation Property	21
ParameterManager Class	21
Fields	22
ConfigFileName Field	22
HTMLParameters Field	23
Constructors	23
ParameterManager Constructor	23
Methods	23
Add Method	23
HandleChangedParameter Method	23
LockMenuItem Method	23
ReadProductionNames Method	23
Refresh Method	23
Remove Method	24
ResetAll Method	24
Properties	24
Enabled Property	24
Events	24
ParameterUpdated Event	24
Parser Class	24
Fields	25

errors Field	25
la Field	25
Productions Field	25
scanner Field	25
t Field	25
Constructors	26
Parser Constructor	26
Methods	26
Parse Method	26
SemErr Method	26
Scanner Class	26
Fields	27
buffer Field	27
Constructors	27
Scanner	27
Methods	28
Peek Method	28
ResetPeek Method	28
Scan Method	28
Properties	28
Line Property	28
Token Class	28
Fields	29
col Field	29
kind Field	29
line Field	29
next Field	29
pos Field	29
val Field	30
<b>Files</b>	<b>30</b>
AbstractPrinterLogic.cs	30
AssemblyInfo.cs	30
CustomPrinterLogic.cs	30
DisplayGrammarControl.cs	30
ExecutablePrinterLogic.cs	31
HTMLForm.cs	31
LibraryPrinterLogic.cs	31
Main.cs	31
MainForm.cs	31
ParameterManager.cs	32

Parser.cs	32
Scanner.cs	32

## Index

**a**


Doc-O-Matic 4 - made with a trial version of Doc-O-Matic 4 - made with a trial version  
- www.doc-o-matic.com - www.doc-o-matic.com - www.doc-o-matic.com - www.doc-o-matic.com - www.doc-o-matic.com

# 1 Symbol Reference


## 1.1 net.stefanhauser.Printy.GUI Namespace

This is namespace net.stefanhauser.Printy.GUI.

### Classes

Class	Description
 AbstractPrinterLogic (see page 2)	Diese Klasse verwaltet die zentrale Steuerung des Printy-Assemblies. Da es sich hierbei um eine abstrakte Superklasse handelt, können in diesem Projekt drei verschiedene Optionen zur Einbindung verwendet werden: Direkter Zugriff auf das Executable über die Kommandozeile, indirekter Zugriff über ein Library und dessen öffentlicher Methode. Teilweise Neuimplementierung des LibrariesIn diesem Projekt wird dafür die zweite Variante, der indirekte Zugriff über das Library, verwendet. Weiters ist es eine Aufgabe dieser Klasse, die Ein- und Ausgabedateien zu verwalten. Dabei wird beim Setzen einer neuen Eingabedatei dessen kompletter Inhalt in eine temporäre Eingabedatei transferiert. Bei jedem Aufruf des Printers wird zusätzlich in eine temporäre Ausgabedatei geschrieben. Erst wenn die Datei gespeichert werden soll, wird der Inhalt der temporären Ausgabedatei in die endgültige Ausgabedatei geschrieben.
Buffer (see page 6)	This is class net.stefanhauser.Printy.GUI.Buffer.
CustomPrinterLogic (see page 8)	Diese Klasse stellt die Möglichkeit dar, die Printer-Logik teilweise nachzuimplementieren. Der Vorteil dieser Möglichkeit ist, dass die Grammatik nur bei Änderung der Eingabedatei neu gelesen werden muss. Zusätzlich muss die formatierte Ausgabe nicht erst in eine Datei geschrieben und im Anschluss wieder gelesen werden. Der Nachteil dieser Möglichkeit ist, dass durch den zusätzlichen kopierten Code eine Inkonsistenz bei der Wartung auftreten kann.
DisplayGrammarControl (see page 10)	Dieses Steuerelement stellt eine Zusammenfassung mehrerer Steuerelemente dar. Darin enthalten sind unter anderem die RichTextBox, in welcher die Grammatik dargestellt wird, die Konsole, in welcher die Parameter dargestellt werden, und den Refresh-Button
Errors (see page 12)	end Parser (see page 24)
ExecutablePrinterLogic (see page 14)	Diese Klasse stellt die Möglichkeit dar, die Printy-Logik über ein Executable zu steuern. Diese Methode wird im Projekt PrintyUI nicht verwendet, da sie einerseits sehr langsam ist und das Executable durch bösen Code leicht ersetzt werden könnte
HTMLForm (see page 16)	Diese Klasse stellt ein Fenster dar, mit dem alle Parameter (see page 17) eingestellt werden können, die mit dem HTML-Output zu tun haben. Dabei handelt es sich um folgende Parameter (see page 17): htmlconfigindexoverviewstyleBis auf den Parameter (see page 17) "html" benötigen alle Parameter (see page 17) eine Dateispezifikation. Um die Parameter (see page 17) für das Interface IAdjustable zusammenzufassen, wird "html" verwendet
LibraryPrinterLogic (see page 17)	Diese Klasse stellt die Möglichkeit dar, das Printy-Assembly als Library zu integrieren. Diese Möglichkeit wird im Projekt PrintyUI verwendet
MainClass (see page 19)	Die Main (see page 20)-Klasse des PrintyUI-Assemblies, welche lediglich die Start-Methode enthält
MainForm (see page 20)	Das Hauptfenster der Printy-Applikation
ParameterManager (see page 21)	Diese Klasse verwaltet alle Parameter, die über die jeweiligen Steuerelemente gesteuert werden können. Dabei müssen diese Steuerelemente das Interface IAdjustable implementieren und sich über entsprechende Methoden registrieren. Sobald sich einer der registrierten Parameter ändert, wird die gesamte Printy-Logik ausgeführt und die Grammatik entsprechend den spezifizierten Parametern formatiert.
Parser (see page 24)	This is class net.stefanhauser.Printy.GUI.Parser.
Scanner (see page 26)	This is class net.stefanhauser.Printy.GUI.Scanner.
Token (see page 28)	This is class net.stefanhauser.Printy.GUI.Token.

## Legend

	abstract
---	----------

## 1.1.1 AbstractPrinterLogic Class

Diese Klasse verwaltet die zentrale Steuerung des Printy-Assemblies. Da es sich hierbei um eine abstrakte Superklasse handelt, können in diesem Projekt drei verschiedene Optionen zur Einbindung verwendet werden: Direkter Zugriff auf das Executable über die Kommandozeile, indirekter Zugriff über ein Library und dessen öffentlicher Methode. Teilweise Neuimplementierung des Libraries. In diesem Projekt wird dafür die zweite Variante, der indirekte Zugriff über das Library, verwendet.

Weiters ist es eine Aufgabe dieser Klasse, die Ein- und Ausgabe Dateien zu verwalten. Dabei wird beim Setzen einer neuen Eingabedatei dessen kompletter Inhalt in eine temporäre Eingabedatei transferiert. Bei jedem Aufruf des Printers wird zusätzlich in eine temporäre Ausgabedatei geschrieben. Erst wenn die Datei gespeichert werden soll, wird der Inhalt der temporären Ausgabedatei in die endgültige Ausgabedatei geschrieben.

### Class Hierarchy


`net.stefanhauser.Printy.GUI.AbstractPrint...`

```
public abstract class AbstractPrinterLogic;
```

### File







AbstractPrinterLogic.cs (see page 30)

### Constructors


Constructor	Description
 <code>AbstractPrinterLogic</code> (see page 4)	Erzeugt eine neue Printer-Logik

### Members







#### Fields

Field	Description
 <code>formattedOutput</code> (see page 3)	Ein String, in dem sich die formatierte Grammatik befindet. Dieser String wird jedesmal nach einer Formatierung aus der temporären Ausgabedatei gefüllt. Der Inhalt des Strings kann dazu verwendet werden, die Grammatik in einer TextBox darzustellen.
 <code>parameters</code> (see page 3)	Die Parameter, mit denen die Grammatik formatiert werden soll.
 <code>tmpInFile</code> (see page 3)	This is tmpInFile, a member of class AbstractPrinterLogic.
 <code>TmpInFile</code> (see page 4)	Der Name der temporären Ein- bzw. Ausgabedateien
 <code>tmpOutFile</code> (see page 4)	This is tmpOutFile, a member of class AbstractPrinterLogic.
 <code>TmpOutFile</code> (see page 4)	Der Name der temporären Ein- bzw. Ausgabedateien





### Constructors

Constructor	Description
 <code>AbstractPrinterLogic</code> (see page 4)	Erzeugt eine neue Printer-Logik









### Methods

Method	Description
 <code>CreateFormattedOutput</code> (see page 4)	Erstellt eine Formatierung der Grammatik.
 <code>DeleteTempFiles</code> (see page 4)	Löscht die temporären Dateien
 <code>FormatOutput</code> (see page 4)	Mit dieser Methode wird die Grammatik formatiert. Das Ergebnis dieser Formatierung wird in die Datei tmpOutFile (see page 4) geschrieben.
 <code>OnInputFileChanged</code> (see page 5)	Wird aufgerufen, sobald sich die Eingabedatei geändert hat.
 <code>OnOutputFileChanged</code> (see page 5)	Wird aufgerufen, sobald sich die Ausgabedatei geändert hat.
 <code>ReadFromInputFile</code> (see page 5)	Liest den Inhalt einer neuen Eingabedatei in die temporäre Eingabedatei




  ReadOutputFromFile (see page 5)	Liest den Inhalt der temporären Ausgabedatei und schreibt diesen in das string-Feld formattedOutput (see page 3). Diese Methode wird nach jeder Formatierung aufgerufen.
  WriteToOutputFile (see page 5)	Schreibt die gesamte formatierte Grammatik in eine Ausgabedatei









## Properties

Property	Description
 CommandLine (see page 6)	Liefert die komplette Kommandozeile in einem String, inklusive dem Namen des Executables, sowie der Ein- als auch der Ausgabedatei.
 FormattedOutput (see page 6)	Liefert die Ausgabe, nachdem eine Grammatik formatiert worden ist.
  InputFile (see page 6)	Setzt oder liefert die Eingabedatei. Wird ein neuer Wert gesetzt, wird der Inhalt dieser Datei unmittelbar in die temporäre Input-Datei übertragen.
 Location (see page 6)	Die Location des Assemblies. Dieser Parameter ist notwendig, um eventuelle Leerzeichen im Pfad zu eliminieren und ausschließlich den Dateinamen selbst in der Konsole darstellen zu können.
  OutputFile (see page 6)	Setzt oder liefert die Ausgabedatei. Wird ein neuer Wert gesetzt, wird der Inhalt der temporären Ausgabedatei sofort in diese Datei übertragen.
 ParameterLine (see page 6)	Liefert alle gesetzten Parameter ausser die Dateinamen in einem String

## Events

Event	Description
 OutputFileChanged (see page 6)	Eventhandler vom Typ FilenameEventHandler, die aufgerufen werden, wenn sich der Wert für die Ein- oder Ausgabedatei ändert

## Legend

	protected
	Data Member
	static
	Method
	abstract
	virtual
	Property
	Event

## 1.1.1.1 Fields

### 1.1.1.1.1 AbstractPrinterLogic.formattedOutput Field

Ein String, in dem sich die formatierte Grammatik befindet. Dieser String wird jedesmal nach einer Formatierung aus der temporären Ausgabedatei gefüllt.

Der Inhalt des Strings kann dazu verwendet werden, die Grammatik in einer TextBox darzustellen.

```
protected string formattedOutput = System.String.Empty;
```

### 1.1.1.1.2 AbstractPrinterLogic.parameters Field

Die Parameter, mit denen die Grammatik formatiert werden soll.

```
protected string[] parameters;
```

### 1.1.1.1.3 AbstractPrinterLogic.tmpInFile Field

```
protected string tmpInFile = TmpInFile;
```

## Remarks

This is tmpInFile, a member of class AbstractPrinterLogic.

#### 1.1.1.1.4 AbstractPrinterLogic.TmpInFile Field

Der Name der temporären Ein- bzw. Ausgabedateien

```
public static string TmpInFile = "in.atg.tmp";
```

#### 1.1.1.1.5 AbstractPrinterLogic.tmpOutFile Field

```
protected string tmpOutFile = TmpOutFile;
```

##### Remarks

This is tmpOutFile, a member of class AbstractPrinterLogic.

#### 1.1.1.1.6 AbstractPrinterLogic.TmpOutFile Field

Der Name der temporären Ein- bzw. Ausgabedateien

```
public static string TmpOutFile = "out.atg.tmp";
```

### 1.1.1.2 Constructors

#### 1.1.1.2.1 AbstractPrinterLogic.AbstractPrinterLogic Constructor

Erzeugt eine neue Printer-Logik

```
public AbstractPrinterLogic();
```

### 1.1.1.3 Methods

#### 1.1.1.3.1 AbstractPrinterLogic.CreateFormattedOutput Method

Erstellt eine Formatierung der Grammatik.

```
public void CreateFormattedOutput(string[] parameters);
```

##### Parameters

Parameters	Description
string[] parameters	Die Parameter zur Formatierung.

#### 1.1.1.3.2 AbstractPrinterLogic.DeleteTempFiles Method

Löscht die temporären Dateien

```
public void DeleteTempFiles(object sender, EventArgs e);
```

#### 1.1.1.3.3 AbstractPrinterLogic.FormatOutput Method

Mit dieser Methode wird die Grammatik formatiert. Das Ergebnis dieser Formatierung wird in die Datei tmpOutFile (see page 4) geschrieben.

```
protected abstract void FormatOutput();
```

### 1.1.1.3.4 AbstractPrinterLogic.OnInputFileChanged Method

Wird aufgerufen, sobald sich die Eingabedatei geändert hat.

```
protected virtual void OnInputFileChanged(object sender, FilenameEventArgs e);
```

#### Parameters

Parameters	Description
newFileName	Der Name der neuen Eingabedatei

### 1.1.1.3.5 AbstractPrinterLogic.OnOutputFileChanged Method

Wird aufgerufen, sobald sich die Ausgabedatei geändert hat.

```
protected virtual void OnOutputFileChanged(object sender, FilenameEventArgs e);
```

#### Parameters

Parameters	Description
newFileName	Der Name der neuen Ausgabedatei

### 1.1.1.3.6 AbstractPrinterLogic.ReadFromInputFile Method

Liest den Inhalt einer neuen Eingabedatei in die temporäre Eingabedatei

```
public void ReadFromInputFile();
```

### 1.1.1.3.7 AbstractPrinterLogic.ReadOutputFromFile Method

Liest den Inhalt der temporären Ausgabedatei und schreibt diesen in das string-Feld formattedOutput (☞ see page 3). Diese Methode wird nach jeder Formatierung aufgerufen.

```
protected void ReadOutputFromFile();
```

### 1.1.1.3.8 WriteToOutputFile

#### 1.1.1.3.8.1 AbstractPrinterLogic.WriteToOutputFile Method ()

Schreibt die gesamte formatierte Grammatik in eine Ausgabedatei

```
public void WriteToOutputFile();
```

#### 1.1.1.3.8.2 AbstractPrinterLogic.WriteToOutputFile Method (string)

Schreibt die gesamte formatierte Grammatik in eine Ausgabedatei

```
public void WriteToOutputFile(string filename);
```

#### Parameters

Parameters	Description
string filename	Der Name der Ausgabedatei

## 1.1.1.4 Properties

#### 1.1.1.4.1 AbstractPrinterLogic.CommandLine Property

Liefert die komplette Kommandozeile in einem String, inklusive dem Namen des Executables, sowie der Ein- als auch der Ausgabedatei.

```
public string CommandLine;
```

#### 1.1.1.4.2 AbstractPrinterLogic.FormattedOutput Property

Liefert die Ausgabe, nachdem eine Grammatik formatiert worden ist.

```
public string FormattedOutput;
```

#### 1.1.1.4.3 AbstractPrinterLogic.InputFile Property

Setzt oder liefert die Eingabedatei. Wird ein neuer Wert gesetzt, wird der Inhalt dieser Datei unmittelbar in die temporäre Input-Datei übertragen.

```
public virtual string InputFile;
```

#### 1.1.1.4.4 AbstractPrinterLogic.Location Property

Die Location des Assemblies. Dieser Parameter ist notwendig, um eventuelle Leerzeichen im Pfad zu eliminieren und ausschließlich den Dateinamen selbst in der Konsole darstellen zu können.

```
public string Location;
```

#### 1.1.1.4.5 AbstractPrinterLogic.OutputFile Property

Setzt oder liefert die Ausgabedatei. Wird ein neuer Wert gesetzt, wird der Inhalt der temporären Ausgabedatei sofort in diese Datei übertragen.

```
public virtual string OutputFile;
```

#### 1.1.1.4.6 AbstractPrinterLogic.ParameterLine Property

Liefert alle gesetzten Parameter ausser die Dateinamen in einem String

```
public string ParameterLine;
```

### 1.1.1.5 Events

#### 1.1.1.5.1 AbstractPrinterLogic.OutputFileChanged Event

Eventhandler vom Typ FilenameEventHandler, die aufgerufen werden, wenn sich der Wert für die Ein- oder Ausgabedatei ändert

```
public event FilenameEventHandler OutputFileChanged;
```

---

## 1.1.2 Buffer Class

### Class Hierarchy

```
net.stefanhauser.Printy.GUI.Buffer
```

```
public class Buffer;
```


## File

Scanner.cs (see page 32)

## Remarks

This is class net.stefanhauser.Printy.GUI.Buffer.

## Constructors

Constructor	Description
 Buffer (see page 7)	This is Buffer, a member of class Buffer.

## Members




### Fields

Field	Description
 EOF (see page 7)	This is EOF, a member of class Buffer.


## Constructors

Constructor	Description
 Buffer (see page 7)	This is Buffer, a member of class Buffer.




## Methods

Method	Description
 GetString (see page 8)	This is GetString, a member of class Buffer.
 Peek (see page 8)	This is Peek, a member of class Buffer.
 Read (see page 8)	This is Read, a member of class Buffer.

## Properties

Property	Description
 Pos (see page 8)	This is Pos, a member of class Buffer.

## Legend

	Data Member
	Method
	Property

## 1.1.2.1 Fields

### 1.1.2.1.1 Buffer.EOF Field

```
public const char EOF = (char)256;
```

## Remarks

This is EOF, a member of class Buffer.

## 1.1.2.2 Constructors

### 1.1.2.2.1 Buffer.Buffer Constructor

```
public Buffer(Stream s, bool isUserStream);
```

**Remarks**

This is Buffer, a member of class Buffer.

## 1.1.2.3 Methods

### 1.1.2.3.1 Buffer.GetString Method

```
public string GetString(int beg, int end);
```

**Remarks**

This is GetString, a member of class Buffer.

### 1.1.2.3.2 Buffer.Peek Method

```
public int Peek();
```

**Remarks**

This is Peek, a member of class Buffer.

### 1.1.2.3.3 Buffer.Read Method

```
public int Read();
```

**Remarks**

This is Read, a member of class Buffer.

## 1.1.2.4 Properties

### 1.1.2.4.1 Buffer.Pos Property

```
public int Pos;
```

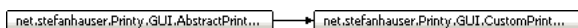
**Remarks**

This is Pos, a member of class Buffer.

---

## 1.1.3 CustomPrinterLogic Class

Diese Klasse stellt die Möglichkeit dar, die Printer-Logik teilweise nachzuimplementieren. Der Vorteil dieser Möglichkeit ist, dass die Grammatik nur bei Änderung der Eingabedatei neu gelesen werden muss. Zusätzlich muss die formatierte Ausgabe nicht erst in eine Datei geschrieben und im Anschluss wieder gelesen werden. Der Nachteil dieser Möglichkeit ist, dass durch den zusätzlichen kopierten Code eine Inkonsistenz bei der Wartung auftreten kann.

**Class Hierarchy**

```
public class CustomPrinterLogic : AbstractPrinterLogic;
```

**File**

CustomPrinterLogic.cs (see page 30)

**Constructors****AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
◆ AbstractPrinterLogic (see page 4)	Erzeugt eine neue Printer-Logik

**Members****Methods**

Method	Description
◆ FormatOutput (see page 10)	public

**AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
◆ CreateFormattedOutput (see page 4)	Erstellt eine Formatierung der Grammatik.
◆ DeleteTempFiles (see page 4)	Löscht die temporären Dateien
◆ FormatOutput (see page 4)	Mit dieser Methode wird die Grammatik formatiert. Das Ergebnis dieser Formatierung wird in die Datei tmpOutFile (see page 4) geschrieben.
◆ OnInputFileChanged (see page 5)	Wird aufgerufen, sobald sich die Eingabedatei geändert hat.
◆ OnOutputFileChanged (see page 5)	Wird aufgerufen, sobald sich die Ausgabedatei geändert hat.
◆ ReadFromInputFile (see page 5)	Liest den Inhalt einer neuen Eingabedatei in die temporäre Eingabedatei
◆ ReadOutputFromFile (see page 5)	Liest den Inhalt der temporären Ausgabedatei und schreibt diesen in das string-Feld formattedOutput (see page 3). Diese Methode wird nach jeder Formatierung aufgerufen.
◆ WriteToOutputFile (see page 5)	Schreibt die gesamte formatierte Grammatik in eine Ausgabedatei

**Properties**

Property	Description
◆ InputFile (see page 10)	Setzt oder liefert die Eingabedatei

**AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
◆ CommandLine (see page 6)	Liefert die komplette Kommandozeile in einem String, inklusive dem Namen des Executables, sowie der Ein- als auch der Ausgabedatei.
◆ FormattedOutput (see page 6)	Liefert die Ausgabe, nachdem eine Grammatik formatiert worden ist.
◆ InputFile (see page 6)	Setzt oder liefert die Eingabedatei. Wird ein neuer Wert gesetzt, wird der Inhalt dieser Datei unmittelbar in die temporäre Input-Datei übertragen.
◆ Location (see page 6)	Die Location des Assemblies. Dieser Parameter ist notwendig, um eventuelle Leerzeichen im Pfad zu eliminieren und ausschließlich den Dateinamen selbst in der Konsole darstellen zu können.
◆ OutputFile (see page 6)	Setzt oder liefert die Ausgabedatei. Wird ein neuer Wert gesetzt, wird der Inhalt der temporären Ausgabedatei sofort in diese Datei übertragen.
◆ ParameterLine (see page 6)	Liefert alle gesetzten Parameter ausser die Dateinamen in einem String


**Fields****AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
◆ formattedOutput (see page 3)	Ein String, in dem sich die formatierte Grammatik befindet. Dieser String wird jedesmal nach einer Formatierung aus der temporären Ausgabedatei gefüllt. Der Inhalt des Strings kann dazu verwendet werden, die Grammatik in einer TextBox darzustellen.
◆ parameters (see page 3)	Die Parameter, mit denen die Grammatik formatiert werden soll.
◆ tmpInFile (see page 3)	This is tmpInFile, a member of class AbstractPrinterLogic.
◆ TmpInFile (see page 4)	Der Name der temporären Ein- bzw. Ausgabedateien
◆ tmpOutFile (see page 4)	This is tmpOutFile, a member of class AbstractPrinterLogic.
◆ TmpOutFile (see page 4)	Der Name der temporären Ein- bzw. Ausgabedateien




## Constructors

### AbstractPrinterLogic Class









AbstractPrinterLogic Class	Description
 AbstractPrinterLogic (see page 4)	Erzeugt eine neue Printer-Logik

## Events

### AbstractPrinterLogic Class

AbstractPrinterLogic Class	Description
 OutputFileChanged (see page 6)	Eventhandler vom Typ FilenameEventHandler, die aufgerufen werden, wenn sich der Wert für die Ein- oder Ausgabedatei ändert

## Legend

	protected
	Method
	abstract
	virtual
	Property
	Data Member
	static
	Event

## 1.1.3.1 Methods

### 1.1.3.1.1 CustomPrinterLogic.FormatOutput Method

```
protected override void FormatOutput();
```

#### Remarks

public

## 1.1.3.2 Properties

### 1.1.3.2.1 CustomPrinterLogic.InputFile Property

Setzt oder liefert die Eingabedatei

```
public override string InputFile;
```

## 1.1.4 DisplayGrammarControl Class

Dieses Steuerelement stellt eine Zusammenfassung mehrerer Steuerelemente dar. Darin enthalten sind unter anderem die RichTextBox, in welcher die Grammatik dargestellt wird, die Konsole, in welcher die Parameter dargestellt werden, und den Refresh-Button

#### Class Hierarchy




```
public class DisplayGrammarControl : System.Windows.Forms.UserControl;
```







**File**

DisplayGrammarControl.cs ([↗](#) see page 30)


**Constructors**

Constructor	Description
 DisplayGrammarControl ( <a href="#">↗</a> see page 12)	Erzeugt ein neues DisplayGrammarControl-Objekt

**Members****Fields**

Field	Description
 console ( <a href="#">↗</a> see page 11)	This is console, a member of class DisplayGrammarControl.
 Gap ( <a href="#">↗</a> see page 11)	Der seitliche (horizontale und vertikale) Mindestabstand aller enthaltenen Steuerelemente zum Rand des Elementes.
 refresh ( <a href="#">↗</a> see page 11)	This is refresh, a member of class DisplayGrammarControl.
 Rtb ( <a href="#">↗</a> see page 12)	This is Rtb, a member of class DisplayGrammarControl.




**Constructors**

Constructor	Description
 DisplayGrammarControl ( <a href="#">↗</a> see page 12)	Erzeugt ein neues DisplayGrammarControl-Objekt

**Methods**

Method	Description
 Dispose ( <a href="#">↗</a> see page 12)	Clean up any resources being used.
 OnSizeChanged ( <a href="#">↗</a> see page 12)	Wird aufgerufen, wenn sich die Größe des Steuerelementes ändert

**Legend**

	Data Member
	Method
	protected

## 1.1.4.1 Fields

### 1.1.4.1.1 DisplayGrammarControl.console Field

```
public System.Windows.Forms.TextBox console;
```

**Remarks**

This is console, a member of class DisplayGrammarControl.

### 1.1.4.1.2 DisplayGrammarControl.Gap Field

Der seitliche (horizontale und vertikale) Mindestabstand aller enthaltenen Steuerelemente zum Rand des Elementes.

```
public int Gap = 8;
```

### 1.1.4.1.3 DisplayGrammarControl.refresh Field

```
public System.Windows.Forms.Button refresh;
```

**Remarks**

This is refresh, a member of class DisplayGrammarControl.

### 1.1.4.1.4 DisplayGrammarControl.Rtb Field

```
public System.Windows.Forms.RichTextBox Rtb;
```

#### Remarks

This is Rtb, a member of class DisplayGrammarControl.

## 1.1.4.2 Constructors

### 1.1.4.2.1 DisplayGrammarControl.DisplayGrammarControl Constructor

Erzeugt ein neues DisplayGrammarControl-Objekt

```
public DisplayGrammarControl();
```

## 1.1.4.3 Methods

### 1.1.4.3.1 DisplayGrammarControl.Dispose Method

Clean up any resources being used.

```
protected override void Dispose(bool disposing);
```

### 1.1.4.3.2 DisplayGrammarControl.OnSizeChanged Method

Wird aufgerufen, wenn sich die Größe des Steuerelementes ändert

```
protected override void OnSizeChanged(EventArgs e);
```

## 1.1.5 Errors Class

#### Class Hierarchy

```
net.stefanhauser.Printy.GUI.Errors
```

```
public class Errors;
```

#### File

Parser.cs (see page 32)

#### Remarks

end Parser (see page 24)

#### Members

##### Fields

Field	Description
count (see page 13)	number of errors detected
errMsgFormat (see page 13)	0=line, 1=column, 2=text

**Methods**

Method	Description
◆ Error (see page 13)	This is Error, a member of class Errors.
◆ Exception (see page 13)	This is Exception, a member of class Errors.
◆ SemErr (see page 13)	This is SemErr, a member of class Errors.
◆ SynErr (see page 14)	This is SynErr, a member of class Errors.

**Legend**

◆	Data Member
◆	Method

**1.1.5.1 Fields****1.1.5.1.1 Errors.count Field**

```
public int count = 0;
```

**Remarks**

number of errors detected

**1.1.5.1.2 Errors.errMsgFormat Field**

```
public string errMsgFormat = "-- line {0} col {1}: {2}";
```

**Remarks**

0=line, 1=column, 2=text

**1.1.5.2 Methods****1.1.5.2.1 Errors.Error Method**

```
public void Error(int line, int col, string s);
```

**Remarks**

This is Error, a member of class Errors.

**1.1.5.2.2 Errors.Exception Method**

```
public void Exception(string s);
```

**Remarks**

This is Exception, a member of class Errors.

**1.1.5.2.3 Errors.SemErr Method**

```
public void SemErr(int line, int col, int n);
```

**Remarks**

This is SemErr, a member of class Errors.

### 1.1.5.2.4 Errors.SynErr Method

```
public void SynErr(int line, int col, int n);
```

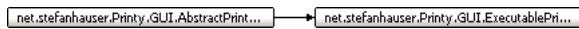
#### Remarks

This is SynErr, a member of class Errors.

## 1.1.6 ExecutablePrinterLogic Class

Diese Klasse stellt die Möglichkeit dar, die Printy-Logik über ein Executable zu steuern. Diese Methode wird im Projekt PrintyUI nicht verwendet, da sie einerseits sehr langsam ist und das Executable durch böartigen Code leicht ersetzt werden könnte

#### Class Hierarchy



```
public class ExecutablePrinterLogic : AbstractPrinterLogic;
```

#### File

ExecutablePrinterLogic.cs (see page 31)

#### Constructors

##### AbstractPrinterLogic Class

AbstractPrinterLogic Class	Description
◆ AbstractPrinterLogic (see page 4)	Erzeugt eine neue Printer-Logik

#### Members

##### Methods

Method	Description
◆ FormatOutput (see page 15)	Formatiert die Ausgabe nach den spezifizierten Parametern

##### AbstractPrinterLogic Class

AbstractPrinterLogic Class	Description
◆ CreateFormattedOutput (see page 4)	Erstellt eine Formatierung der Grammatik.
◆ DeleteTempFiles (see page 4)	Löscht die temporären Dateien
◆ FormatOutput (see page 4)	Mit dieser Methode wird die Grammatik formatiert. Das Ergebnis dieser Formatierung wird in die Datei tmpOutFile (see page 4) geschrieben.
◆ OnInputFileChanged (see page 5)	Wird aufgerufen, sobald sich die Eingabedatei geändert hat.
◆ OnOutputFileChanged (see page 5)	Wird aufgerufen, sobald sich die Ausgabedatei geändert hat.
◆ ReadFromInputFile (see page 5)	Liest den Inhalt einer neuen Eingabedatei in die temporäre Eingabedatei
◆ ReadOutputFromFile (see page 5)	Liest den Inhalt der temporären Ausgabedatei und schreibt diesen in das string-Feld formattedOutput (see page 3). Diese Methode wird nach jeder Formatierung aufgerufen.
◆ WriteToOutputFile (see page 5)	Schreibt die gesamte formatierte Grammatik in eine Ausgabedatei

#### Properties

Property	Description
◆ ApplicationName (see page 16)	Setzt oder liefert den Namen des Executables ohne Suffix

##### AbstractPrinterLogic Class

AbstractPrinterLogic Class	Description
◆ CommandLine (see page 6)	Liefert die komplette Kommandozeile in einem String, inklusive dem Namen des Executables, sowie der Ein- als auch der Ausgabedatei.

FormattedOutput (see page 6)	Liefert die Ausgabe, nachdem eine Grammatik formatiert worden ist.
InputFile (see page 6)	Setzt oder liefert die Eingabedatei. Wird ein neuer Wert gesetzt, wird der Inhalt dieser Datei unmittelbar in die temporäre Input-Datei übertragen.
Location (see page 6)	Die Location des Assemblies. Dieser Parameter ist notwendig, um eventuelle Leerzeichen im Pfad zu eliminieren und ausschließlich den Dateinamen selbst in der Konsole darstellen zu können.
OutputFile (see page 6)	Setzt oder liefert die Ausgabedatei. Wird ein neuer Wert gesetzt, wird der Inhalt der temporären Ausgabedatei sofort in diese Datei übertragen.
ParameterLine (see page 6)	Liefert alle gesetzten Parameter ausser die Dateinamen in einem String

**Fields**

**AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
formattedOutput (see page 3)	Ein String, in dem sich die formatierte Grammatik befindet. Dieser String wird jedesmal nach einer Formatierung aus der temporären Ausgabedatei gefüllt. Der Inhalt des Strings kann dazu verwendet werden, die Grammatik in einer TextBox darzustellen.
parameters (see page 3)	Die Parameter, mit denen die Grammatik formatiert werden soll.
tmpInFile (see page 3)	This is tmpInFile, a member of class AbstractPrinterLogic.
TmplnFile (see page 4)	Der Name der temporären Ein- bzw. Ausgabedateien
tmpOutFile (see page 4)	This is tmpOutFile, a member of class AbstractPrinterLogic.
TmpOutFile (see page 4)	Der Name der temporären Ein- bzw. Ausgabedateien

**Constructors**

**AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
AbstractPrinterLogic (see page 4)	Erzeugt eine neue Printer-Logik

**Events**

**AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
OutputFileChanged (see page 6)	Eventhandler vom Typ FilenameEventHandler, die aufgerufen werden, wenn sich der Wert für die Ein- oder Ausgabedatei ändert

**Legend**

	protected
	Method
	abstract
	virtual
	Property
	Data Member
	static
	Event

**1.1.6.1 Methods**

**1.1.6.1.1 ExecutablePrinterLogic.FormatOutput Method**

Formatiert die Ausgabe nach den spezifizierten Parametern

```
protected override void FormatOutput();
```

**1.1.6.2 Properties**

Doc-O-Matic 4 - made with a trial version of Doc-O-Matic 4 - www.doc-o-matic.com

### 1.1.6.2.1 ExecutablePrinterLogic.ApplicationName Property

Setzt oder liefert den Namen des Executables ohne Suffix

```
public string ApplicationName;
```

## 1.1.7 HTMLForm Class

Diese Klasse stellt ein Fenster dar, mit dem alle Parameter (☒ see page 17) eingestellt werden können, die mit dem HTML-Output zu tun haben. Dabei handelt es sich um folgende Parameter (☒ see page 17): htmlconfigindexoverviewstyleBis auf den Parameter (☒ see page 17) "html" benötigen alle Parameter (☒ see page 17) eine Dateispezifikation. Um die Parameter (☒ see page 17) für das Interface IAdjustable zusammenzufassen, wird "html" verwendet

### Class Hierarchy



```
public class HTMLForm : Form, IAdjustable;
```

### File

HTMLForm.cs (☒ see page 31)

### Constructors

Constructor	Description
☒ HTMLForm (☒ see page 17)	Erzeugt ein neues HTML-Form-Objekt

### Members

### Constructors

Constructor	Description
☒ HTMLForm (☒ see page 17)	Erzeugt ein neues HTML-Form-Objekt

### Methods

Method	Description
☒ Dispose (☒ see page 17)	Clean up any resources being used.
☒ Reset (☒ see page 17)	Setzt alle Steuerelemente wieder auf Default
☒ SetParameterEventHandler (☒ see page 17)	Setzt den Listener für das IAdjustable-Element
☒ SetupHelpProvider (☒ see page 17)	This is SetupHelpProvider, a member of class HTMLForm.

### Properties

Property	Description
☒ AdjustedParameter (☒ see page 17)	Liefert den angepassten Parameter (☒ see page 17)
☒ Parameter (☒ see page 17)	Liefert den Parameter

### Legend

☒	Method
☒	protected
☒	Property

### 1.1.7.1 Constructors

### 1.1.7.1.1 HTMLForm.HTMLForm Constructor

Erzeugt ein neues HTML-Form-Objekt

```
public HTMLForm();
```

### 1.1.7.2 Methods

#### 1.1.7.2.1 HTMLForm.Dispose Method

Clean up any resources being used.

```
protected override void Dispose(bool disposing);
```

#### 1.1.7.2.2 HTMLForm.Reset Method

Setzt alle Steuerelemente wieder auf Default

```
public void Reset();
```

#### 1.1.7.2.3 HTMLForm.SetParameterEventHandler Method

Setzt den Listener für das IAdjustable-Element

```
public void SetParameterEventHandler(ParameterEventHandler listener);
```

#### 1.1.7.2.4 HTMLForm.SetupHelpProvider Method

```
public void SetupHelpProvider(HelpProvider provider);
```

#### Remarks

This is SetupHelpProvider, a member of class HTMLForm.

### 1.1.7.3 Properties

#### 1.1.7.3.1 HTMLForm.AdjustedParameter Property

Liefert den angepassten Parameter (☞ see page 17)

```
public string AdjustedParameter;
```

#### 1.1.7.3.2 HTMLForm.Parameter Property

Liefert den Parameter

```
public string Parameter;
```

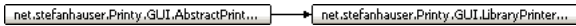
---

## 1.1.8 LibraryPrinterLogic Class

Diese Klasse stellt die Möglichkeit dar, das Printy-Assembly als Library zu integrieren. Diese Möglichkeit wird im Projekt PrintyUI verwendet



**Class Hierarchy**



```
public class LibraryPrinterLogic : AbstractPrinterLogic;
```

**File**

LibraryPrinterLogic.cs (see page 31)

**Constructors**

**AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
◆ AbstractPrinterLogic (see page 4)	Erzeugt eine neue Printer-Logik

**Members**

**Methods**

Method	Description
◆ FormatOutput (see page 19)	Liefert eine formatierte Ausgabe

**AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
◆ CreateFormattedOutput (see page 4)	Erstellt eine Formatierung der Grammatik.
◆ DeleteTempFiles (see page 4)	Löscht die temporären Dateien
◆ FormatOutput (see page 4)	Mit dieser Methode wird die Grammatik formatiert. Das Ergebnis dieser Formatierung wird in die Datei tmpOutFile (see page 4) geschrieben.
◆ OnInputFileChanged (see page 5)	Wird aufgerufen, sobald sich die Eingabedatei geändert hat.
◆ OnOutputFileChanged (see page 5)	Wird aufgerufen, sobald sich die Ausgabedatei geändert hat.
◆ ReadFromInputFile (see page 5)	Liest den Inhalt einer neuen Eingabedatei in die temporäre Eingabedatei
◆ ReadOutputFromFile (see page 5)	Liest den Inhalt der temporären Ausgabedatei und schreibt diesen in das string-Feld formattedOutput (see page 3). Diese Methode wird nach jeder Formatierung aufgerufen.
◆ WriteToOutputFile (see page 5)	Schreibt die gesamte formatierte Grammatik in eine Ausgabedatei

**Fields**

**AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
◆ formattedOutput (see page 3)	Ein String, in dem sich die formatierte Grammatik befindet. Dieser String wird jedesmal nach einer Formatierung aus der temporären Ausgabedatei gefüllt. Der Inhalt des Strings kann dazu verwendet werden, die Grammatik in einer TextBox darzustellen.
◆ parameters (see page 3)	Die Parameter, mit denen die Grammatik formatiert werden soll.
◆ tmpInFile (see page 3)	This is tmpInFile, a member of class AbstractPrinterLogic.
◆ TmpInFile (see page 4)	Der Name der temporären Ein- bzw. Ausgabedateien
◆ tmpOutFile (see page 4)	This is tmpOutFile, a member of class AbstractPrinterLogic.
◆ TmpOutFile (see page 4)	Der Name der temporären Ein- bzw. Ausgabedateien

**Constructors**

**AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
◆ AbstractPrinterLogic (see page 4)	Erzeugt eine neue Printer-Logik







**Properties**

**AbstractPrinterLogic Class**

AbstractPrinterLogic Class	Description
◆ CommandLine (see page 6)	Liefert die komplette Kommandozeile in einem String, inklusive dem Namen des Executables, sowie der Ein- als auch der Ausgabedatei.
◆ FormattedOutput (see page 6)	Liefert die Ausgabe, nachdem eine Grammatik formatiert worden ist.


Doc-O-Matic 4 - made with a trial version of Doc-O-Matic 4 - www.doc-o-matic.com - made with a trial version of Doc-O-Matic 4 - www.doc-o-matic.com











  InputFile (see page 6)	Setzt oder liefert die Eingabedatei. Wird ein neuer Wert gesetzt, wird der Inhalt dieser Datei unmittelbar in die temporäre Input-Datei übertragen.
 Location (see page 6)	Die Location des Assemblies. Dieser Parameter ist notwendig, um eventuelle Leerzeichen im Pfad zu eliminieren und ausschließlich den Dateinamen selbst in der Konsole darstellen zu können.
  OutputFile (see page 6)	Setzt oder liefert die Ausgabedatei. Wird ein neuer Wert gesetzt, wird der Inhalt der temporären Ausgabedatei sofort in diese Datei übertragen.
 ParameterLine (see page 6)	Liefert alle gesetzten Parameter ausser die Dateinamen in einem String

## Events

### AbstractPrinterLogic Class

AbstractPrinterLogic Class	Description
 OutputFileChanged (see page 6)	Eventhandler vom Typ FilenameEventHandler, die aufgerufen werden, wenn sich der Wert für die Ein- oder Ausgabedatei ändert

## Legend

	protected
	Method
	abstract
	virtual
	Data Member
	static
	Property
	Event

## 1.1.8.1 Methods

### 1.1.8.1.1 LibraryPrinterLogic.FormatOutput Method

Liefert eine formatierte Ausgabe

```
protected override void FormatOutput();
```

## 1.1.9 MainClass Class

Die Main (see page 20)-Klasse des PrintyUI-Assemblies, welche lediglich die Start-Methode enthält

### Class Hierarchy

```
net.stefanhauser.Printy.GUI.MainClass
```



```
public class MainClass;
```

### File



Main.cs (see page 31)

### Members

### Methods

Method	Description
  Main (see page 20)	Die Main-Methode des Assemblies

## Legend

	Method
	static

## 1.1.9.1 Methods

### 1.1.9.1.1 MainClass.Main Method

Die Main-Methode des Assemblies

```
[STAThread]
public static void Main(string[] args);
```

## 1.1.10 MainForm Class

Das Hauptfenster der Printy-Applikation

### Class Hierarchy




```
public class MainForm : System.Windows.Forms.Form;
```

### File


MainForm.cs (see page 31)

### Constructors






Constructor	Description
 MainForm (see page 21)	This is MainForm, a member of class MainForm.

### Members


#### Constructors

Constructor	Description
 MainForm (see page 21)	This is MainForm, a member of class MainForm.





#### Methods

Method	Description
   InitializeCustomComponents (see page 21)	This is InitializeCustomComponents, a member of class MainForm.
  OnLoad (see page 21)	Wird aufgerufen, wenn die Applikation geladen wird. Wurde im Konstruktor ein Dateiname spezifiziert, wird der Inhalt dieser Datei geparkt und in der RichTextBox dargestellt.

#### Properties

Property	Description
 AssemblyLocation (see page 21)	Die Location des Assemblies

#### Legend

	Method
	protected
	virtual
	Property

## 1.1.10.1 Constructors

### 1.1.10.1.1 MainForm

### 1.1.10.1.1 MainForm.MainForm Constructor ()

```
public MainForm();
```

#### Remarks

This is MainForm, a member of class MainForm.

### 1.1.10.1.2 MainForm.MainForm Constructor (string)

Erzeugt ein neues MainForm-Objekt

```
public MainForm(string filename);
```

#### Parameters

Parameters	Description
string filename	Der Name der Datei, die formatiert werden soll

## 1.1.10.2 Methods

### 1.1.10.2.1 MainForm.InitializeCustomComponents Method

```
protected virtual void InitializeCustomComponents();
```

#### Remarks

This is InitializeCustomComponents, a member of class MainForm.

### 1.1.10.2.2 MainForm.OnLoad Method

Wird aufgerufen, wenn die Applikation geladen wird. Wurde im Konstruktor ein Dateiname spezifiziert, wird der Inhalt dieser Datei geparkt und in der RichTextBox dargestellt.

```
protected override void OnLoad(EventArgs e);
```

## 1.1.10.3 Properties

### 1.1.10.3.1 MainForm.AssemblyLocation Property

Die Location des Assemblies

```
public string AssemblyLocation;
```

## 1.1.11 ParameterManager Class

Diese Klasse verwaltet alle Parameter, die über die jeweiligen Steuerelemente gesteuert werden können. Dabei müssen diese Steuerelemente das Interface IAdjustable implementieren und sich über entsprechende Methoden registrieren. Sobald sich einer der registrierten Parameter ändert, wird die gesamte Printy-Logik ausgeführt und die Grammatik entsprechend den spezifizierten Parametern formatiert.

#### Class Hierarchy

```
net.stefanhauser.Printy.GUI.ParameterM...
```

```
public class ParameterManager;
```

**File**

ParameterManager.cs (🔗 see page 32)

**Constructors**

Constructor	Description
🔗 ParameterManager (🔗 see page 23)	Erzeugt ein neues ParameterManager-Objekt

**Members****Fields**

Field	Description
🔗 ConfigFileName (🔗 see page 22)	Die Parameter des Config-Fensters. Dient als Zwischenablage
🔗 HTMLParameters (🔗 see page 23)	Die Parameter des HTML-Fensters. Dient als Zwischenablage

**Constructors**

Constructor	Description
🔗 ParameterManager (🔗 see page 23)	Erzeugt ein neues ParameterManager-Objekt

**Methods**

Method	Description
🔗 Add (🔗 see page 23)	Fügt ein neues IAdjustable-Element ein. Sollte dieses Element zusätzlich das Interface IProductionNames implementieren, wird dieses für die Darstellung der Produktionsnamen extra registriert.
🔗 HandleChangedParameter (🔗 see page 23)	Diese Methode wird jedem IAdjustable-Element als Delegate übergeben, das sich im ParameterManager registriert. Dabei werden die gesammelten Parameter neu organisiert und im Anschluss wird die Ausgabe neu formatiert
🔗 LockMenuItem (🔗 see page 23)	Diese Methode fügt alle Menüpunkte dem Manager hinzu, welche bei Programmstart deaktiviert sein sollen. Sobald eine Datei geladen wird, werden all diese Elemente aktiviert.
🔗 ReadProductionNames (🔗 see page 23)	Liest die Namen der Produktionen, sobald eine neue Eingabedatei spezifiziert worden ist. Im Anschluss werden die Produktionsnamen an das registrierte Steuerelement übergeben.
🔗 Refresh (🔗 see page 23)	Wird ausgeführt, wenn der Refresh-Button geklickt wird. Die Ausgabe wird dabei mit den zuletzt gespeicherten Parametern formatiert.
🔗 Remove (🔗 see page 24)	Entfernt ein IAdjustable-Element
🔗 ResetAll (🔗 see page 24)	Bringt alle registrierten Elemente wieder in ihren Ausgangszustand

**Properties**

Property	Description
🔗 Enabled (🔗 see page 24)	Aktiviert oder deaktiviert alle registrierten Elemente

**Events**

Event	Description
⚡ ParameterUpdated (🔗 see page 24)	Wird aufgerufen, sobald sich einer der Parameter geändert hat

**Legend**

🔗	Data Member
🔗	Method
🔗	Property
⚡	Event

## 1.1.11.1 Fields

### 1.1.11.1.1 ParameterManager.ConfigFileName Field

Die Parameter des Config-Fensters. Dient als Zwischenablage

```
public string ConfigFileName = null;
```

### 1.1.11.1.2 ParameterManager.HTMLParameters Field

Die Parameter des HTML-Fensters. Dient als Zwischenablage

```
public string[] HTMLParameters = null;
```

## 1.1.11.2 Constructors

### 1.1.11.2.1 ParameterManager.ParameterManager Constructor

Erzeugt ein neues ParameterManager-Objekt

```
public ParameterManager();
```

## 1.1.11.3 Methods

### 1.1.11.3.1 ParameterManager.Add Method

Fügt ein neues IAdjustable-Element ein. Sollte dieses Element zusätzlich das Interface IProductionNames implementieren, wird dieses für die Darstellung der Produktionsnamen extra registriert.

```
public void Add(IAdjustable element);
```

### 1.1.11.3.2 ParameterManager.HandleChangedParameter Method

Diese Methode wird jedem IAdjustable-Element als Delegate übergeben, das sich im ParameterManager (☞ see page 21) registriert. Dabei werden die gesammelten Parameter neu organisiert und im Anschluss wird die Ausgabe neu formatiert

```
public void HandleChangedParameter(IAdjustable sender, EventArgs e);
```

### 1.1.11.3.3 ParameterManager.LockMenuItem Method

Diese Methode fügt alle Menüpunkte dem Manager hinzu, welche bei Programmstart deaktiviert sein sollen. Sobald eine Datei geladen wird, werden all diese Elemente aktiviert.

```
public void LockMenuItem(System.Windows.Forms.MenuItem c);
```

#### Parameters

Parameters	Description
System.Windows.Forms.MenuItem c	Steuerelement

### 1.1.11.3.4 ParameterManager.ReadProductionNames Method

Liest die Namen der Produktionen, sobald eine neue Eingabedatei spezifiziert worden ist. Im Anschluss werden die Produktionsnamen an das registrierte Steuerelement übergeben.

```
public void ReadProductionNames(object sender, FilenameEventArgs e);
```

### 1.1.11.3.5 ParameterManager.Refresh Method

Wird ausgeführt, wenn der Refresh-Button geklickt wird. Die Ausgabe wird dabei mit den zuletzt gespeicherten Parametern formatiert.

```
public void Refresh(object sender, EventArgs e);
```

### 1.1.11.3.6 ParameterManager.Remove Method

Entfernt ein IAdjustable-Element

```
public void Remove(IAdjustable element);
```

### 1.1.11.3.7 ParameterManager.ResetAll Method

Bringt alle registrierten Elemente wieder in ihren Ausgangszustand

```
public void ResetAll();
```

## 1.1.11.4 Properties

### 1.1.11.4.1 ParameterManager.Enabled Property

Aktiviert oder deaktiviert alle registrierten Elemente

```
public bool Enabled;
```

## 1.1.11.5 Events

### 1.1.11.5.1 ParameterManager.ParameterUpdated Event

Wird aufgerufen, sobald sich einer der Parameter geändert hat

```
public event AdjustedParameterEventHandler ParameterUpdated;
```

## 1.1.12 Parser Class

### Class Hierarchy

```
net.stefanhauser.Printy.GUI.Parser
```

```
public class Parser;
```

### File

Parser.cs (see page 32)

### Remarks

This is class net.stefanhauser.Printy.GUI.Parser.

### Constructors




Constructor	Description
 Parser (see page 26)	This is Parser, a member of class Parser.

### Members


#### Fields

Field	Description
 errors (see page 25)	This is errors, a member of class Parser.
 la (see page 25)	lookahead token

Created with a demo version of **Doc-O-Matic**. This version is supplied for evaluation purposes only, do not distribute this documentation. To obtain a commercial license please see <http://www.doc-o-matic.com/purchase.html>.

 Productions (see page 25)	This is Productions, a member of class Parser.
 scanner (see page 25)	This is scanner, a member of class Parser.
 t (see page 25)	last recognized token



## Constructors

Constructor	Description
 Parser (see page 26)	This is Parser, a member of class Parser.

## Methods

Method	Description
 Parse (see page 26)	This is Parse, a member of class Parser.
 SemErr (see page 26)	This is SemErr, a member of class Parser.

## Legend

	Data Member
	Method

## 1.1.12.1 Fields

### 1.1.12.1.1 Parser.errors Field

```
public Errors errors;
```

#### Remarks

This is errors, a member of class Parser.

### 1.1.12.1.2 Parser.la Field

```
public Token la;
```

#### Remarks

lookahead token

### 1.1.12.1.3 Parser.Productions Field

```
public ArrayList Productions;
```

#### Remarks

This is Productions, a member of class Parser.

### 1.1.12.1.4 Parser.scanner Field

```
public Scanner scanner;
```

#### Remarks

This is scanner, a member of class Parser.

### 1.1.12.1.5 Parser.t Field

```
public Token t;
```

**Remarks**

last recognized token

## 1.1.12.2 Constructors

### 1.1.12.2.1 Parser.Parser Constructor

```
public Parser(Scanner scanner);
```

**Remarks**

This is Parser, a member of class Parser.

## 1.1.12.3 Methods

### 1.1.12.3.1 Parser.Parse Method

```
public void Parse();
```

**Remarks**

This is Parse, a member of class Parser.

### 1.1.12.3.2 Parser.SemErr Method

```
public void SemErr(string msg);
```

**Remarks**

This is SemErr, a member of class Parser.

## 1.1.13 Scanner Class

**Class Hierarchy**

```
net.stefanhauser.Printy.GUI.Scanner
```

```
public class Scanner;
```

**File**

Scanner.cs (see page 32)

**Remarks**


This is class net.stefanhauser.Printy.GUI.Scanner.

**Constructors**

Constructor	Description
Scanner (see page 27)	This is Scanner, a member of class Scanner.






**Members****Fields**

Field	Description
 buffer (see page 27)	scanner buffer


**Constructors**

Constructor	Description
 Scanner (see page 27)	This is Scanner, a member of class Scanner.





**Methods**

Method	Description
 Peek (see page 28)	peek for the next token, ignore pragmas
 ResetPeek (see page 28)	make sure that peeking starts at the current scan position
 Scan (see page 28)	get the next token (possibly a token already seen during peeking)

**Properties**

Property	Description
 Line (see page 28)	PG 2005-07-04 Gibt die aktuelle Zeile des Scanners zurück; Dies wird gebraucht, um mehrzeilige Kommentare effizienter zu erkennen

**Legend**

	Data Member
	Method
	internal
	Property

## 1.1.13.1 Fields

### 1.1.13.1.1 Scanner.buffer Field

```
public Buffer buffer;
```

**Remarks**

scanner buffer

## 1.1.13.2 Constructors

### 1.1.13.2.1 Scanner

#### 1.1.13.2.1.1 Scanner.Scanner Constructor (Stream)

```
public Scanner(Stream s);
```

**Remarks**

This is Scanner, a member of class Scanner.

#### 1.1.13.2.1.2 Scanner.Scanner Constructor (string)

```
public Scanner(string fileName);
```

**Remarks**

This is Scanner, a member of class Scanner.

## 1.1.13.3 Methods

### 1.1.13.3.1 Scanner.Peek Method

```
public Token Peek();
```

**Remarks**

peek for the next token, ignore pragmas

### 1.1.13.3.2 Scanner.ResetPeek Method

```
public void ResetPeek();
```

**Remarks**

make sure that peeking starts at the current scan position

### 1.1.13.3.3 Scanner.Scan Method

```
public Token Scan();
```

**Remarks**

get the next token (possibly a token already seen during peeking)

## 1.1.13.4 Properties

### 1.1.13.4.1 Scanner.Line Property

PG 2005-07-04 Gibt die aktuelle Zeile des Scanners zurÃ¼ck; Dies wird gebraucht, um mehrzeilige Kommentare effizienter zu erkennen

```
internal int Line;
```

---

## 1.1.14 Token Class

**Class Hierarchy**

```
net.stefanhauser.Printy.GUI.Token
```

```
public class Token;
```







**File**

Scanner.cs (see page 32)


**Remarks**

This is class net.stefanhauser.Printy.GUI.Token.

**Members****Fields**

Field	Description
 col (see page 29)	token column (starting at 0)
 kind (see page 29)	token kind
 line (see page 29)	token line (starting at 1)
 next (see page 29)	ML 2005-03-11 Tokens are kept in linked list
 pos (see page 29)	token position in the source text (starting at 0)
 val (see page 30)	token value

**Legend**

	Data Member
---	-------------

## 1.1.14.1 Fields

### 1.1.14.1.1 Token.col Field

```
public int col;
```

**Remarks**

token column (starting at 0)

### 1.1.14.1.2 Token.kind Field

```
public int kind;
```

**Remarks**

token kind

### 1.1.14.1.3 Token.line Field

```
public int line;
```

**Remarks**

token line (starting at 1)

### 1.1.14.1.4 Token.next Field

```
public Token next;
```

**Remarks**

ML 2005-03-11 Tokens are kept in linked list

### 1.1.14.1.5 Token.pos Field

```
public int pos;
```

**Remarks**

token position in the source text (starting at 0)

### 1.1.14.1.6 Token.val Field

```
public string val;
```

#### Remarks

token value

## 1.2 Files

### 1.2.1 AbstractPrinterLogic.cs

This is file AbstractPrinterLogic.cs.

#### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (see page 1)	This is namespace net.stefanhauser.Printy.GUI.

### 1.2.2 AssemblyInfo.cs

This is file AssemblyInfo.cs.

### 1.2.3 CustomPrinterLogic.cs

This is file CustomPrinterLogic.cs.

#### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (see page 1)	This is namespace net.stefanhauser.Printy.GUI.

### 1.2.4 DisplayGrammarControl.cs

This is file DisplayGrammarControl.cs.

#### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (see page 1)	This is namespace net.stefanhauser.Printy.GUI.

---

## 1.2.5 ExecutablePrinterLogic.cs

This is file ExecutablePrinterLogic.cs.

### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (☒ see page 1)	This is namespace net.stefanhauser.Printy.GUI.

---

## 1.2.6 HTMLForm.cs

This is file HTMLForm.cs.

### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (☒ see page 1)	This is namespace net.stefanhauser.Printy.GUI.

---

## 1.2.7 LibraryPrinterLogic.cs

This is file LibraryPrinterLogic.cs.

### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (☒ see page 1)	This is namespace net.stefanhauser.Printy.GUI.

---

## 1.2.8 Main.cs

This is file Main.cs.

### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (☒ see page 1)	This is namespace net.stefanhauser.Printy.GUI.

---

## 1.2.9 MainForm.cs

This is file MainForm.cs.

### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (☒ see page 1)	This is namespace net.stefanhauser.Printy.GUI.

---

## 1.2.10 ParameterManager.cs

This is file ParameterManager.cs.

### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (see page 1)	This is namespace net.stefanhauser.Printy.GUI.

---

## 1.2.11 Parser.cs

This is file Parser.cs.

### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (see page 1)	This is namespace net.stefanhauser.Printy.GUI.

---

## 1.2.12 Scanner.cs

This is file Scanner.cs.

### Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI (see page 1)	This is namespace net.stefanhauser.Printy.GUI.

# Index

## A

AbstractPrinterLogic class 2  
     AbstractPrinterLogic 4  
     CommandLine 6  
     CreateFormattedOutput 4  
     DeleteTempFiles 4  
     FormatOutput 4  
     formattedOutput 3  
     FormattedOutput 6  
     InputFile 6  
     Location 6  
     OnInputFileChanged 5  
     OnOutputFileChanged 5  
     OutputFile 6  
     OutputFileChanged 6  
     ParameterLine 6  
     parameters 3  
     ReadFromInputFile 5  
     ReadOutputFromFile 5  
     tmpInFile 3  
     TmpInFile 4  
     tmpOutFile 4  
     TmpOutFile 4  
     WriteToOutputFile 5  
 AbstractPrinterLogic.cs 20  
 AssemblyInfo.cs 30

## B

Buffer class 6  
     Buffer 7  
     EOF 7  
     GetString 8  
     Peek 8  
     Pos 8  
     Read 8

## C

CustomPrinterLogic class 8  
     FormatOutput 10  
     InputFile 10  
 CustomPrinterLogic.cs 30

## D

DisplayGrammarControl class 10  
     console 11  
     DisplayGrammarControl 12  
     Dispose 12  
     Gap 11  
     OnSizeChanged 12  
     refresh 11  
     Rtb 12  
 DisplayGrammarControl.cs 30

## E

Errors class 12  
     count 13  
     errMsgFormat 13  
     Error 13  
     Exception 13  
     SemErr 13  
     SynErr 14  
 ExecutablePrinterLogic class 14  
     ApplicationName 16  
     FormatOutput 15  
 ExecutablePrinterLogic.cs 31

## H

HTMLForm class 16  
     AdjustedParameter 17  
     Dispose 17  
     HTMLForm 17  
     Parameter 17  
     Reset 17  
     SetParameterEventHandler 17

SetupHelpProvider 17	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.Location 6
HTMLForm.cs 31	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.OnInputFile Changed 5
<b>L</b>	
LibraryPrinterLogic class 17	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.OnOutputFil eChanged 5
FormatOutput 19	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.OutputFile 6
LibraryPrinterLogic.cs 31	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.OutputFileC hanged 6
<b>M</b>	
Main.cs 31	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.ParameterLi ne 6
MainClass class 19	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.parameters 3
Main 20	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.ReadFromIn putFile 5
MainForm class 20	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.ReadOutput FromFile 5
AssemblyLocation 21	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.tmpInFile 3
InitializeCustomComponents 21	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.TmpInFile 4
MainForm 21	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.tmpOutFile 4
OnLoad 21	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.TmpOutFile 4
MainForm.cs 31	net.stefanhauser.Printy.GUI.AbstractPrinterLogic.WriteToOutp utFile 5
<b>N</b>	
net.stefanhauser.Printy.GUI 1	net.stefanhauser.Printy.GUI.Buffer 6
net.stefanhauser.Printy.GUI namespace 1	net.stefanhauser.Printy.GUI.Buffer.Buffer 7
net.stefanhauser.Printy.GUI.AbstractPrinterLogic 2	net.stefanhauser.Printy.GUI.Buffer.EOF 7
net.stefanhauser.Printy.GUI.AbstractPrinterLogic.AbstractPrint erLogic 4	net.stefanhauser.Printy.GUI.Buffer.GetString 8
net.stefanhauser.Printy.GUI.AbstractPrinterLogic.CommandLi ne 6	net.stefanhauser.Printy.GUI.Buffer.Peek 8
net.stefanhauser.Printy.GUI.AbstractPrinterLogic.CreateForm attedOutput 4	net.stefanhauser.Printy.GUI.Buffer.Pos 8
net.stefanhauser.Printy.GUI.AbstractPrinterLogic.DeleteTemp Files 4	net.stefanhauser.Printy.GUI.Buffer.Read 8
net.stefanhauser.Printy.GUI.AbstractPrinterLogic.FormatOutp ut 4	net.stefanhauser.Printy.GUI.CustomPrinterLogic 8
net.stefanhauser.Printy.GUI.AbstractPrinterLogic.formattedOu tput 3	net.stefanhauser.Printy.GUI.CustomPrinterLogic.FormatOutpu t 10
net.stefanhauser.Printy.GUI.AbstractPrinterLogic.FormattedO utput 6	net.stefanhauser.Printy.GUI.CustomPrinterLogic.InputFile 10
net.stefanhauser.Printy.GUI.AbstractPrinterLogic.InputFile 6	net.stefanhauser.Printy.GUI.DisplayGrammarControl 10
	net.stefanhauser.Printy.GUI.DisplayGrammarControl.console 11
	net.stefanhauser.Printy.GUI.DisplayGrammarControl.DisplayG rammarControl 12



net.stefanhauser.Printy.GUI.DisplayGrammarControl.Dispose 12	net.stefanhauser.Printy.GUI.MainForm.MainForm 21
net.stefanhauser.Printy.GUI.DisplayGrammarControl.Gap 11	net.stefanhauser.Printy.GUI.MainForm.OnLoad 21
net.stefanhauser.Printy.GUI.DisplayGrammarControl.OnSizeChanged 12	net.stefanhauser.Printy.GUI.ParameterManager 21
net.stefanhauser.Printy.GUI.DisplayGrammarControl.refresh 11	net.stefanhauser.Printy.GUI.ParameterManager.Add 23
net.stefanhauser.Printy.GUI.DisplayGrammarControl.Rtb 12	net.stefanhauser.Printy.GUI.ParameterManager.ConfigFileName 22
net.stefanhauser.Printy.GUI.Errors 12	net.stefanhauser.Printy.GUI.ParameterManager.Enabled 24
net.stefanhauser.Printy.GUI.Errors.count 13	net.stefanhauser.Printy.GUI.ParameterManager.HandleChangedParameter 23
net.stefanhauser.Printy.GUI.Errors.errMsgFormat 13	net.stefanhauser.Printy.GUI.ParameterManager.HTMLParameters 23
net.stefanhauser.Printy.GUI.Errors.Error 13	net.stefanhauser.Printy.GUI.ParameterManager.LockMenuItem 23
net.stefanhauser.Printy.GUI.Errors.Exception 13	net.stefanhauser.Printy.GUI.ParameterManager.ParameterManager 23
net.stefanhauser.Printy.GUI.Errors.SemErr 13	net.stefanhauser.Printy.GUI.ParameterManager.ParameterUpdated 24
net.stefanhauser.Printy.GUI.Errors.SynErr 14	net.stefanhauser.Printy.GUI.ParameterManager.ReadProductionNames 23
net.stefanhauser.Printy.GUI.ExecutablePrinterLogic 14	net.stefanhauser.Printy.GUI.ParameterManager.Refresh 23
net.stefanhauser.Printy.GUI.ExecutablePrinterLogic.ApplicationName 16	net.stefanhauser.Printy.GUI.ParameterManager.Remove 24
net.stefanhauser.Printy.GUI.ExecutablePrinterLogic.FormatOutput 15	net.stefanhauser.Printy.GUI.ParameterManager.ResetAll 24
net.stefanhauser.Printy.GUI.HTMLForm 16	net.stefanhauser.Printy.GUI.Parser 24
net.stefanhauser.Printy.GUI.HTMLForm.AdjustedParameter 17	net.stefanhauser.Printy.GUI.Parser.errors 25
net.stefanhauser.Printy.GUI.HTMLForm.Dispose 17	net.stefanhauser.Printy.GUI.Parser.la 25
net.stefanhauser.Printy.GUI.HTMLForm.HTMLForm 17	net.stefanhauser.Printy.GUI.Parser.Parse 26
net.stefanhauser.Printy.GUI.HTMLForm.Parameter 17	net.stefanhauser.Printy.GUI.Parser.Parser 26
net.stefanhauser.Printy.GUI.HTMLForm.Reset 17	net.stefanhauser.Printy.GUI.Parser.Productions 25
net.stefanhauser.Printy.GUI.HTMLForm.SetParameterEventHandler 17	net.stefanhauser.Printy.GUI.Parser.scanner 25
net.stefanhauser.Printy.GUI.HTMLForm.SetupHelpProvider 17	net.stefanhauser.Printy.GUI.Parser.SemErr 26
net.stefanhauser.Printy.GUI.LibraryPrinterLogic 17	net.stefanhauser.Printy.GUI.Parser.t 25
net.stefanhauser.Printy.GUI.LibraryPrinterLogic.FormatOutput 19	net.stefanhauser.Printy.GUI.Scanner 26
net.stefanhauser.Printy.GUI.MainClass 19	net.stefanhauser.Printy.GUI.Scanner.buffer 27
net.stefanhauser.Printy.GUI.MainClass.Main 20	net.stefanhauser.Printy.GUI.Scanner.Line 28
net.stefanhauser.Printy.GUI.MainForm 20	net.stefanhauser.Printy.GUI.Scanner.Peek 28
net.stefanhauser.Printy.GUI.MainForm.AssemblyLocation 21	net.stefanhauser.Printy.GUI.Scanner.ResetPeek 28
net.stefanhauser.Printy.GUI.MainForm.InitializeCustomComponents 21	net.stefanhauser.Printy.GUI.Scanner.Scan 28
	net.stefanhauser.Printy.GUI.Scanner.Scanner 27
	net.stefanhauser.Printy.GUI.Token 28

net.stefanhauser.Printy.GUI.Token.col 29  
 net.stefanhauser.Printy.GUI.Token.kind 29  
 net.stefanhauser.Printy.GUI.Token.line 29  
 net.stefanhauser.Printy.GUI.Token.next 29  
 net.stefanhauser.Printy.GUI.Token.pos 29  
 net.stefanhauser.Printy.GUI.Token.val 30

## P

ParameterManager class 21

Add 23

ConfigFileName 22

Enabled 24

HandleChangedParameter 23

HTMLParameters 23

LockMenuItem 23

ParameterManager 23

ParameterUpdated 24

ReadProductionNames 23

Refresh 23

Remove 24

ResetAll 24

ParameterManager.cs 32

Parser class 24

errors 25

la 25

Parse 26

Parser 26

Productions 25

scanner 25

SemErr 26

t 25

Parser.cs 32

## S

Scanner class 26

buffer 27

Line 28

Peek 28

ResetPeek 28

Scan 28

Scanner 27

Scanner.cs 32

## T

Token class 28

col 29

kind 29

line 29

next 29

pos 29

val 30