

Printy Config Namespace

Table of Contents

Symbol Reference

net.stefanhauser.Printy.GUI.Config Namespace	1
AbstractRuleControl Class	1
Fields	2
groupBox1 Field	3
ruleType Field	3
Methods	4
GetCreatedRule Method	4
SetupHelpProvider Method	4
Properties	4
Text Property	4
BitSetControl Class	4
Constructors	5
BitSetControl Constructor	6
Methods	6
Dispose Method	6
GetCreatedRule Method	6
BreakRuleControl Class	6
Constructors	7
BreakRuleControl Constructor	7
Methods	7
Dispose Method	7
GetCreatedRule Method	8
Buffer Class	8
Fields	8
EOF Field	9
Constructors	9
Buffer Constructor	9
Methods	9
Close Method	9
GetString Method	9
Peek Method	9
Read Method	9
Properties	9
Pos Property	9
ClassControl Class	9

Constructors	10
ClassControl Constructor	11
Methods	11
Dispose Method	11
GetCreatedRule Method	11
ConfigForm Class	11
Constructors	12
ConfigForm Constructor	12
Methods	12
Dispose Method	12
Reset Method	12
SetParameterEventHandler Method	12
SetupHelpProvider Method	13
Properties	13
AdjustedParameter Property	13
ConfigFileName Property	13
Parameter Property	13
Errors Class	13
Fields	14
count Field	14
errMsgFormat Field	14
Constructors	15
Errors Constructor	15
Methods	15
CreateException Method	15
Error Method	15
Exception Method	15
SemErr Method	15
SynErr Method	16
Properties	16
ErrorCount Property	16
Parser Class	16
Fields	19
errors Field	20
la Field	20
Rules Field	20
scanner Field	20
t Field	20
Constructors	20
Parser Constructor	20

Methods	20
Parse Method	21
SemErr Method	21
ParserError Class	21
Constructors	22
ParserError Constructor	22
Methods	22
ToString Method	22
Properties	22
Column Property	22
Line Property	22
Text Property	22
Type Property	22
ParserException Class	23
Constructors	23
ParserException Constructor	23
Properties	23
ErrorCount Property	23
ErrorList Property	24
Rule Class	24
Fields	25
BreakMode Field	25
FirstSymbol Field	25
FurtherText Field	25
Number Field	25
SecondSymbol Field	25
Text Field	25
Type Field	26
Methods	26
Add	26
CreateBitSet Method	26
Equals	26
IsValid Method	27
ToString Method	27
RuleControl Class	27
Fields	28
ActiveControl Field	28
Constructors	28
RuleControl Constructor	28
Methods	28

Dispose Method	39
GetCreatedRule Method	39
Token Class	39
Fields	40
col Field	40
kind Field	40
line Field	40
next Field	40
pos Field	41
val Field	41
Utilities Class	41
Properties	41
BreakRules Property	41
Rules Property	42
SingleSymbol Property	42
SymbolAll Property	42
SymbolAllIndent Property	42
Wrapper Class	42
Fields	43
BreakMode Field	43
RuleType Field	43
Symbol Field	43
Constructors	43
Wrapper	43
Methods	44
ToString Method	44
Operators	44
BreakMode Operator	44
RuleType Operator	45
Symbol Operator	45
IRuleControl Interface	45
Methods	45
GetCreatedRule Method	45
Properties	45
Text Property	46
BreakMode Enumeration	46
ErrorType Enumeration	46
RuleType Enumeration	47
Symbol Enumeration	47
RuleEventHandler Type	48

Files	49
AbstractRuleControl.cs	49
AssemblyInfo.cs	49
BitSetControl.cs	49
BreakRuleControl.cs	50
ClassControl.cs	50
ConfigForm.cs	50
Enumerations.cs	51
Parser.cs	51
Rule.cs	52
RuleControl.cs	52
RuleControlFactory.cs	52
RuleManager.cs	53
Scanner.cs	53
SpaceRuleControl.cs	54
Utilities.cs	54
Wrapper.cs	54

Index **a**

1 Symbol Reference

1.1 net.stefanhauser.Printy.GUI.Config Namespace


Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet

```
-config <filename>
```

In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.

Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.

Classes

Class	Description
 AbstractRuleControl (see page 2)	Dieses abstrakte Steuerelement stellt die Basisklasse jedes Steuerelementes dar, welches zur Erstellung einer Formatierungsregel dient. In dieser Klasse kann der Typ der Regel mit einem Wert aus der Enumeration RuleType (see page 47) gesetzt werden; gleichzeitig wird dieser Wert in einer GroupBox angezeigt, welche in jedem dieser Elemente enthalten ist.
BitSetControl (see page 4)	Erzeugt ein neues Steuerelement, dass Regeln mit folgendem Format erzeugen kann: Rule (see page 24) = <rulename> Symbol (see page 47); Rule (see page 24) = <rulename> "[" {Symbol (see page 47)} "];
BreakRuleControl (see page 6)	Dieses Steuerelement kann neue Regeln vom Typ BreakRule erzeugen.
Buffer (see page 8)	Der Buffer des Scanners
ClassControl (see page 9)	Dieses Steuerelement kann neue Regeln vom Typ ClassRule erzeugen
ConfigForm (see page 11)	Diese Klasse repräsentiert das Hauptfenster dieser Bibliothek. Dieses Fenster enthält verschiedenste Steuerelemente, die es ermöglichen, verschiedene Formatierungsregeln für den PrettyPrinter zu erstellen.
Errors (see page 13)	Diese Klasse verwaltet alle Fehler, die während des Parsens eines Dokumentes gefunden worden sind. Dabei wird strikt unterschieden zwischen syntaktischen und semantischen Fehlern. Alle Fehler werden während des Parsens in einer Liste gespeichert. Sollte die Liste nach dem Parsen die Liste nicht leer sein, wird eine entsprechende Exception (see page 15) geworfen.
Parser (see page 16)	Dieser Parser wird zum Lesen einer Konfigurationsdatei verwendet. Er wurde mit dem Werkzeug COCO erstellt.
ParserError (see page 21)	Ein Objekt dieser Klasse wird erzeugt, wenn im Parser (see page 16) ein Fehler festgestellt wird.
ParserException (see page 23)	Eine solche Exception wird geworfen, wenn während des Parse-Vorgangs mindestens ein Fehler gefunden worden ist. Diese Exception wird von der Methode Parser.Parse (see page 21)() geworfen.
Rule (see page 24)	Ein Objekt dieser Klasse stellt eine Regel dar, wie sie über graphische Steuerelemente erstellt werden kann. Eine solche Regel hat einen Regeltyp und verschiedene Parameter. Sollte einer dieser Parameter

RuleControl (🔗 see page 27)	Dieses Steuerelement ermöglicht dem Benutzer, Regeln aller Art zu erstellen. Es enthält eine ComboBox, welche alle Regeltypen auflistet. Sobald ein Regeltyp gewählt worden ist, wird das entsprechende Steuerelement angezeigt und in einer entsprechenden Variablen-Instanz gespeichert.
RuleControlFactory (🔗 see page 29)	This is class net.stefanhauser.Printy.GUI.Config.RuleControlFactory.
RuleEventArgs (🔗 see page 29)	Ein Objekt dieser Klasse wird erzeugt, wenn im RuleManager (🔗 see page 30) entweder eine neue Regel hinzugefügt oder entfernt worden ist.
RuleManager (🔗 see page 30)	Diese Klasse verwaltet die Regeln, die über die graphische Benutzeroberfläche erstellt worden sind. Zur Speicherung der Rule (🔗 see page 24)-Objekte wird eine Collection vom Typ ArrayList verwendet. Weiters gibt es in dieser Klasse noch die Möglichkeit, den Namen der Datei zu spezifizieren, aus welcher die Regeln gelesen werden bz. in welche die Regeln geschrieben werden.
Scanner (🔗 see page 33)	Der Scanner, der zum Lesen der Regel-Datei verwendet wird. Diese Klasse wurde durch das Tool COCO erstellt.
SpaceRuleControl (🔗 see page 38)	Mit diesem Steuerelement ist es möglich, neue Regeln vom Typ SpaceRule zu erstellen
Token (🔗 see page 39)	Token-Objekte werden vom Scanner (🔗 see page 33) während des Lesevorgangs geliefert
Utilities (🔗 see page 41)	Diese Klasse enthält verschiedene Sammlungen von Enumationswerten, wie sie im Parser (🔗 see page 16) verwendet und gelesen werden.
Wrapper (🔗 see page 42)	Diese Klasse dient in erster Linie dazu, dass Enumerationen vom Typ RuleType (🔗 see page 43), Symbol (🔗 see page 43) und BreakMode (🔗 see page 43) auch als Objekte verwendet werden können. Verschiedene Steuerelemente wie zum Beispiel eine ListBox akzeptieren für darzustellende Items lediglich object[]-Elemente. Um die Verwendung dieser Klasse auch innerhalb des SourceCodes so weit wie möglich zu verstecken, gibt es in dieser Klasse zahlreiche Methoden zur impliziten und expliziten Typumwandlung.

Enumerations

Enumeration	Description
BreakMode (🔗 see page 46)	Eine Enumeration, die alle Werte enthält, die eine BreakRule annehmen kann
ErrorType (🔗 see page 46)	Diese Enumeration enthält alle Fehlertypen, welche während des Parsens auftreten können.
RuleType (🔗 see page 47)	Diese Enumeration stellt eine Sammlung aller Typen von Regeln dar, mit welchen die Ausgabe des PrettyPrinters konfiguriert werden können.
Symbol (🔗 see page 47)	Diese Enumeration enthält alle Symbole, welche in einer Konfigurationsdatei verwendet werden. Diese Symbole können zum Beispiel dafür verwendet werden, um eine Regel zu erstellen, mit der der Abstand zwischen zwei solchen Symbolen festgelegt werden kann.


Interfaces

Interface	Description
IRuleControl (🔗 see page 45)	Dieses Interface wird von jedem Steuerelement implementiert, welches in der Lage ist, eine Regel zu erstellen.

Types

Type	Description
RuleEventHandler (🔗 see page 48)	Dieses Delegate wird verwendet, wenn eine Regel zum RuleManager (🔗 see page 30) hinzugefügt oder vom RuleManager (🔗 see page 30) entfernt worden ist.

Legend

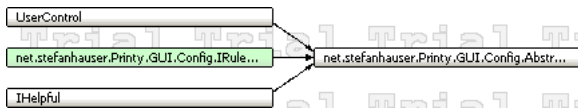
	abstract
---	----------

1.1.1 AbstractRuleControl Class

Dieses abstrakte Steuerelement stellt die Basisklasse jedes Steuerelementes dar, welches zur Erstellung einer Formatierungsregel dient. In dieser Klasse kann der Typ der Regel mit einem Wert aus der Enumeration RuleType (🔗 see page 47) gesetzt werden; gleichzeitig wird dieser Wert in einer GroupBox angezeigt, welche in jedem dieser Elemente

enthalten ist.

Class Hierarchy



```
public abstract class AbstractRuleControl : UserControl, IRuleControl, IHelpful;
```

File

AbstractRuleControl.cs (see page 49)

Members

Fields

Field	Description
groupBox1 (see page 3)	Die GroupBox, welche alle Steuerelemente enthält, mit deren Hilfe man Regeln erstellen kann
ruleType (see page 4)	Der Typ der zu erstellenden Regel

Methods

Method	Description
GetCreatedRule (see page 4)	Diese Methode liefert das Rule (see page 24)-Objekt, welches mit Hilfe der verschiedenen Steuerelemente erstellt worden ist. Diese Methode ist als abstrakt deklariert und muss in jeder erbenden Klasse implementiert werden.
SetupHelpProvider (see page 4)	This is SetupHelpProvider, a member of class AbstractRuleControl.

IRuleControl Interface

IRuleControl Interface	Description
GetCreatedRule (see page 45)	Liefert die erstellte Regel

Properties

Property	Description
Text (see page 4)	Setzt oder liefert den Namen der Regel, die im Steuerelement erstellt werden kann

IRuleControl Interface

IRuleControl Interface	Description
Text (see page 46)	Setzt oder liefert den Typ der Regel

Legend

	protected
	Data Member
	Method
	abstract
	Property

1.1.1.1 Fields

1.1.1.1.1 AbstractRuleControl.groupBox1 Field

```
protected System.Windows.Forms.GroupBox groupBox1;
```

Remarks

Die GroupBox, welche alle Steuerelemente enthält, mit deren Hilfe man Regeln erstellen kann

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

1.1.1.1.2 AbstractRuleControl.ruleType Field

```
protected RuleType ruleType = RuleType.UNSET;
```

Remarks

Der Typ der zu erstellenden Regel

1.1.1.2 Methods

1.1.1.2.1 AbstractRuleControl.GetCreatedRule Method

```
public abstract Rule GetCreatedRule();
```

Remarks

Diese Methode liefert das Rule (see page 24)-Objekt, welches mit Hilfe der verschiedenen Steuerelemente erstellt worden ist. Diese Methode ist als abstrakt deklariert und muss in jeder erbenden Klasse implementiert werden.

1.1.1.2.2 AbstractRuleControl.SetupHelpProvider Method

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

Remarks

This is SetupHelpProvider, a member of class AbstractRuleControl.

1.1.1.3 Properties

1.1.1.3.1 AbstractRuleControl.Text Property

```
public new RuleType Text;
```

Remarks

Setzt oder liefert den Namen der Regel, die im Steuerelement erstellt werden kann

1.1.2 BitSetControl Class

Erzeugt ein neues Steuerelement, dass Regeln mit folgendem Format erzeugen kann:

Rule (see page 24) = <rulename> Symbol (see page 47);

Rule (see page 24) = <rulename> "[" {Symbol (see page 47)} "];"

Class Hierarchy





```
public class BitSetControl : AbstractRuleControl;
```

File



BitSetControl.cs (see page 49)

Constructors


Constructor	Description
  BitSetControl (see page 6)	Erzeugt ein neues BitSetControl-Objekt

Members






Constructors

Constructor	Description
  BitSetControl (see page 6)	Erzeugt ein neues BitSetControl-Objekt

Methods

Method	Description
  Dispose (see page 6)	Clean up any resources being used.
  GetCreatedRule (see page 6)	Liefert das erstellte Rule (see page 24)-Objekt

AbstractRuleControl Class

AbstractRuleControl Class	Description
   GetCreatedRule (see page 4)	Diese Methode liefert das Rule (see page 24)-Objekt, welches mit Hilfe der verschiedenen Steuerelemente erstellt worden ist. Diese Methode ist als abstrakt deklariert und muss in jeder erbenden Klasse implementiert werden.
  SetupHelpProvider (see page 4)	This is SetupHelpProvider, a member of class AbstractRuleControl.

IRuleControl Interface

IRuleControl Interface	Description
  GetCreatedRule (see page 45)	Liefert die erstellte Regel


Fields

AbstractRuleControl Class

AbstractRuleControl Class	Description
 groupBox1 (see page 3)	Die GroupBox, welche alle Steuerelemente enthält, mit deren Hilfe man Regeln erstellen kann
 ruleType (see page 4)	Der Typ der zu erstellenden Regel

Properties






AbstractRuleControl Class

AbstractRuleControl Class	Description
 Text (see page 4)	Setzt oder liefert den Namen der Regel, die im Steuerelement erstellt werden kann

IRuleControl Interface

IRuleControl Interface	Description
 Text (see page 46)	Setzt oder liefert den Typ der Regel

Legend

	Method
	protected
	abstract
	Data Member
	Property

1.1.2.1 Constructors

1.1.2.1.1 BitSetControl.BitSetControl Constructor

```
public BitSetControl();
```

Remarks

Erzeugt ein neues BitSetControl-Objekt

1.1.2.2 Methods

1.1.2.2.1 BitSetControl.Dispose Method

Clean up any resources being used.

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

1.1.2.2.2 BitSetControl.GetCreatedRule Method

```
public override Rule GetCreatedRule();
```

Remarks

Liefert das erstellte Rule (see page 24)-Objekt

1.1.3 BreakRuleControl Class

Dieses Steuerelement kann neue Regeln vom Typ BreakRule erzeugen.

Class Hierarchy



```
public class BreakRuleControl : AbstractRuleControl;
```

File

BreakRuleControl.cs (see page 50)

Constructors

Constructor	Description
◆ BreakRuleControl (see page 7)	Erzeugt ein neues BreakRule-Steuerelement

Members




Constructors

Constructor	Description
◆ BreakRuleControl (see page 7)	Erzeugt ein neues BreakRule-Steuerelement


Methods

Method	Description
◆ Dispose (see page 7)	Clean up any resources being used.
◆ GetCreatedRule (see page 8)	Liefert die erzeugte Break-Regel

AbstractRuleControl Class

AbstractRuleControl Class	Description
  GetCreatedRule (see page 4)	Diese Methode liefert das Rule (see page 24)-Objekt, welches mit Hilfe der verschiedenen Steuerelemente erstellt worden ist. Diese Methode ist als abstrakt deklariert und muss in jeder erbenden Klasse implementiert werden.
 SetupHelpProvider (see page 4)	This is SetupHelpProvider, a member of class AbstractRuleControl.

IRuleControl Interface

IRuleControl Interface	Description
 GetCreatedRule (see page 45)	Liefert die erstellte Regel

Fields

AbstractRuleControl Class

AbstractRuleControl Class	Description
 groupBox1 (see page 3)	Die GroupBox, welche alle Steuerelemente enthält, mit deren Hilfe man Regeln erstellen kann
 ruleType (see page 4)	Der Typ der zu erstellenden Regel

Properties






AbstractRuleControl Class

AbstractRuleControl Class	Description
 Text (see page 4)	Setzt oder liefert den Namen der Regel, die im Steuerelement erstellt werden kann

IRuleControl Interface

IRuleControl Interface	Description
 Text (see page 46)	Setzt oder liefert den Typ der Regel

Legend

	Method
	protected
	abstract
	Data Member
	Property

1.1.3.1 Constructors

1.1.3.1.1 BreakRuleControl.BreakRuleControl Constructor

```
public BreakRuleControl();
```

Remarks

Erzeugt ein neues BreakRule-Steuerelement

1.1.3.2 Methods

1.1.3.2.1 BreakRuleControl.Dispose Method

Clean up any resources being used.

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

1.1.3.2.2 BreakRuleControl.GetCreatedRule Method

```
public override Rule GetCreatedRule();
```

Remarks

Liefert die erzeugte Break-Regel

1.1.4 Buffer Class

Class Hierarchy

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

```
public class Buffer;
```

File

Scanner.cs (see page 53)

Remarks

Der Buffer des Scanners

Constructors

Constructor	Description
 Buffer (see page 9)	

Members





Fields

Field	Description
 EOF (see page 9)	

Constructors

Constructor	Description
 Buffer (see page 9)	




Methods

Method	Description
 Close (see page 9)	This is Close, a member of class Buffer.
 GetString (see page 9)	
 Peek (see page 9)	
 Read (see page 9)	

Properties

Property	Description
 Pos (see page 9)	

Legend

	Data Member
	Method
	Property

1.1.4.1 Fields

1.1.4.1.1 Buffer.EOF Field

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

1.1.4.2 Constructors

1.1.4.2.1 Buffer.Buffer Constructor

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

1.1.4.3 Methods

1.1.4.3.1 Buffer.Close Method

```
public void Close();
```

Remarks

This is Close, a member of class Buffer.

1.1.4.3.2 Buffer.GetString Method

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

1.1.4.3.3 Buffer.Peek Method

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

1.1.4.3.4 Buffer.Read Method

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

1.1.4.4 Properties

1.1.4.4.1 Buffer.Pos Property

```
public int Pos;
```

1.1.5 ClassControl Class

Dieses Steuerelement kann neue Regeln vom Typ ClassRule erzeugen

Class Hierarchy




```
public class ClassControl : AbstractRuleControl;
```

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>.


File

ClassControl.cs (see page 50)

Constructors

Constructor	Description
 ClassControl (see page 11)	Erzeugt ein neues ClassRule-Steuerelement




Members**Constructors**

Constructor	Description
 ClassControl (see page 11)	Erzeugt ein neues ClassRule-Steuerelement

Methods

Method	Description
  Dispose (see page 11)	Clean up any resources being used.
 GetCreatedRule (see page 11)	Liefert die erzeugte Regel vom Typ ClassRule



AbstractRuleControl Class

AbstractRuleControl Class	Description
  GetCreatedRule (see page 4)	Diese Methode liefert das Rule (see page 24)-Objekt, welches mit Hilfe der verschiedenen Steuerelemente erstellt worden ist. Diese Methode ist als abstrakt deklariert und muss in jeder erbenden Klasse implementiert werden.
 SetupHelpProvider (see page 4)	This is SetupHelpProvider, a member of class AbstractRuleControl.


IRuleControl Interface

IRuleControl Interface	Description
 GetCreatedRule (see page 45)	Liefert die erstellte Regel

Fields**AbstractRuleControl Class**

AbstractRuleControl Class	Description
 groupBox1 (see page 3)	Die GroupBox, welche alle Steuerelemente enthält, mit deren Hilfe man Regeln erstellen kann
 ruleType (see page 4)	Der Typ der zu erstellenden Regel






Properties**AbstractRuleControl Class**

AbstractRuleControl Class	Description
 Text (see page 4)	Setzt oder liefert den Namen der Regel, die im Steuerelement erstellt werden kann

IRuleControl Interface

IRuleControl Interface	Description
 Text (see page 46)	Setzt oder liefert den Typ der Regel

Legend

	Method
	protected
	abstract
	Data Member
	Property

1.1.5.1 Constructors

1.1.5.1.1 ClassControl.ClassControl Constructor

```
public ClassControl();
```

Remarks

Erzeugt ein neues ClassRule-Steuerelement

1.1.5.2 Methods

1.1.5.2.1 ClassControl.Dispose Method

Clean up any resources being used.

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

1.1.5.2.2 ClassControl.GetCreatedRule Method

```
public override Rule GetCreatedRule();
```

Remarks

Liefert die erzeugte Regel vom Typ ClassRule

1.1.6 ConfigForm Class

Class Hierarchy



```
public class ConfigForm : System.Windows.Forms.Form,
net.stefanhauser.Printy.GUI.Util.IAdjustable;
```

File

ConfigForm.cs (🔗 see page 50)

Remarks


Diese Klasse repräsentiert das Hauptfenster dieser Bibliothek. Dieses Fenster enthält verschiedenste Steuerelemente, die es ermöglichen, verschiedene Formatierungsregeln für den PrettyPrinter zu erstellen.

- Mit Elementen im oberen Bereich des Fensters kann eingestellt werden, ob eine Formatierungsdatei verwendet werden soll oder nicht.
- Im mittleren Bereich können verschiedene Regeln erstellt, editiert oder gelöscht werden.
- Im unteren Bereich können die vorgenommenen Änderungen übernommen werden und in die spezifizierte Datei geschrieben werden.





Constructors

Constructor	Description
🔗 ConfigForm (🔗 see page 12)	Erzeugt ein neues Fenster




Members**Constructors**

Constructor	Description
 ConfigForm (see page 12)	Erzeugt ein neues Fenster




Methods

Method	Description
 Dispose (see page 12)	Entfernt sämtliche verwendeten Komponenten aus dem Speicher
 Reset (see page 12)	Diese Methode resettet alle in ihr vorhandenen Steuerelemente
 SetParameterEventHandler (see page 12)	Setzt den Listener, der im Hauptprogramm registriert werden muss.
 SetupHelpProvider (see page 13)	This is SetupHelpProvider, a member of class ConfigForm.

Properties

Property	Description
 AdjustedParameter (see page 13)	Dieses Property liefert ein string[]-Objekt, falls die Regel "-config" verwendet werden soll, ansonsten liefert es null.
 ConfigFileName (see page 13)	This is ConfigFileName, a member of class ConfigForm.
 Parameter (see page 13)	Liefert den Namen des Parameters "config"

Legend

	Method
	protected
	Property

1.1.6.1 Constructors

1.1.6.1.1 ConfigForm.ConfigForm Constructor

```
public ConfigForm();
```

Remarks

Erzeugt ein neues Fenster

1.1.6.2 Methods

1.1.6.2.1 ConfigForm.Dispose Method

Entfernt sämtliche verwendeten Komponenten aus dem Speicher

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

1.1.6.2.2 ConfigForm.Reset Method

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

Remarks

Diese Methode resettet alle in ihr vorhandenen Steuerelemente

1.1.6.2.3 ConfigForm.SetParameterEventHandler Method

```
public void SetParameterEventHandler(net.stefanhauser.Printy.GUI.Util.ParameterEventHandler
```

```
listener);
```

Remarks

Setzt den Listener, der im Hauptprogramm registriert werden muss.

1.1.6.2.4 ConfigForm.SetupHelpProvider Method

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

Remarks

This is SetupHelpProvider, a member of class ConfigForm.

1.1.6.3 Properties

1.1.6.3.1 ConfigForm.AdjustedParameter Property

```
public string AdjustedParameter;
```

Remarks

Dieses Property liefert ein string[]-Objekt, falls die Regel "-config" verwendet werden soll, ansonsten liefert es null.

1.1.6.3.2 ConfigForm.ConfigFileName Property

```
public string ConfigFileName;
```

Remarks

This is ConfigFileName, a member of class ConfigForm.

1.1.6.3.3 ConfigForm.Parameter Property

```
public string Parameter;
```

Remarks

Liefert den Namen des Parameters "config"

1.1.7 Errors Class

Class Hierarchy

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

```
public class Errors;
```


File

Parser.cs (see page 51)



Remarks

Diese Klasse verwaltet alle Fehler, die während des Parsens eines Dokumentes gefunden worden sind. Dabei wird strikt unterschieden zwischen syntaktischen und semantischen Fehlern. Alle Fehler werden während des Parsens in einer Liste gespeichert. Sollte die Liste nach dem Parsen die Liste nicht leer sein, wird eine entsprechende Exception (see page 15) geworfen.


Constructors

Constructor	Description
 Errors (see page 15)	Die Liste, in welcher alle aufgetretenen Fehler gespeichert werden.






Members**Fields**

Field	Description
 count (see page 14)	number of errors detected
 errMsgFormat (see page 14)	0=line, 1=column, 2=text


Constructors

Constructor	Description
 Errors (see page 15)	Die Liste, in welcher alle aufgetretenen Fehler gespeichert werden.




Methods

Method	Description
 CreateException (see page 15)	Erstellt die Exception (see page 15), welche die Liste mit allen aufgetretenen Fehlern enthält.
 Error (see page 15)	Diese Methode wird aufgerufen, wenn im Dokument ein allgemeiner Fehler gefunden worden ist. Dabei wird die Position des Fehlers in einem ParserError (see page 21)-Objekt gespeichert und in einer Liste abgelegt.
 Exception (see page 15)	Diese Methode wird vom Parser (see page 16) aufgerufen, wenn ein schwerwiegender Fehler, also eine Exception, während des Parse-Vorgangs auftritt. Dabei wird eine entsprechende Fehlermeldung auf der Konsole ausgegeben und das Programm wird beendet.
 SemErr (see page 15)	Diese Methode wird aufgerufen, wenn im Dokument ein semantischer Fehler gefunden worden ist. Dabei wird die Position des Fehlers in einem ParserError (see page 21)-Objekt gespeichert und in einer Liste abgelegt.
 SynErr (see page 16)	Diese Methode wird aufgerufen, wenn im Dokument ein syntaktischer Fehler gefunden worden ist. Dabei wird die Position des Fehlers in einem ParserError (see page 21)-Objekt gespeichert und in einer Liste abgelegt.

Properties

Property	Description
 ErrorCount (see page 16)	Gibt an, wie viele Fehler, egal welcher Art, im geparsten Dokument gefunden worden sind.

Legend

	Data Member
	Method
	Property

1.1.7.1 Fields**1.1.7.1.1 Errors.count Field**

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

Remarks

number of errors detected

1.1.7.1.2 Errors.errMsgFormat Field

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

Remarks

0=line, 1=column, 2=text

1.1.7.2 Constructors

1.1.7.2.1 Errors.Errors Constructor

```
public Errors();
```

Remarks

Die Liste, in welcher alle aufgetretenen Fehler gespeichert werden.

1.1.7.3 Methods

1.1.7.3.1 Errors.CreateException Method

```
public ParseException CreateException();
```

Remarks

Erstellt die Exception (☞ see page 15), welche die Liste mit allen aufgetretenen Fehlern enthält.

1.1.7.3.2 Errors.Error Method

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

Remarks

Diese Methode wird aufgerufen, wenn im Dokument ein allgemeiner Fehler gefunden worden ist. Dabei wird die Position des Fehlers in einem ParserError (☞ see page 21)-Objekt gespeichert und in einer Liste abgelegt.

1.1.7.3.3 Errors.Exception Method

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

Remarks

Diese Methode wird vom Parser (☞ see page 16) aufgerufen, wenn ein schwerwiegender Fehler, also eine Exception, während des Parse-Vorgangs auftritt. Dabei wird eine entsprechende Fehlermeldung auf der Konsole ausgegeben und das Programm wird beendet.

1.1.7.3.4 Errors.SemErr Method

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

Remarks

Diese Methode wird aufgerufen, wenn im Dokument ein semantischer Fehler gefunden worden ist. Dabei wird die Position des Fehlers in einem ParserError (☞ see page 21)-Objekt gespeichert und in einer Liste abgelegt.

1.1.7.3.5 Errors.SynErr Method

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

Remarks

Diese Methode wird aufgerufen, wenn im Dokument ein syntaktischer Fehler gefunden worden ist. Dabei wird die Position des Fehlers in einem ParserError (see page 21)-Objekt gespeichert und in einer Liste abgelegt.

1.1.7.4 Properties

1.1.7.4.1 Errors.ErrorCount Property

```
public int ErrorCount;
```

Remarks

Gibt an, wie viele Fehler, egal welcher Art, im geparsten Dokument gefunden worden sind.

1.1.8 Parser Class

Class Hierarchy

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

```
public class Parser;
```

File

Parser.cs (see page 51)

Remarks

Dieser Parser wird zum Lesen einer Konfigurationsdatei verwendet. Er wurde mit dem Werkzeug COCO erstellt.

```
/*-----
Config.atg
2005-11 Stefan Hauser

Original Version by Philip Grasböck
-----*/

COMPILER PrintyConfig

CHARACTERS
digit      = "0123456789".
tab        = '\t'.
cr         = '\r'.
lf         = '\n'.
eol        = cr + lf.

TOKENS
number = ['-'] digit { digit }.

COMMENTS FROM "/*" TO "*/" NESTED
COMMENTS FROM "//" TO cr lf

IGNORE eol + tab
```

```

/*-----*/
PRODUCTIONS

PrintyConfig
=
{
    Rule<r>          (. Rule r = new Rule();          .)
    'i'            (. Rules.Add(r);                .)
}
.

Rule<Rule r> = ConfigRule<r> | HyperConfigRule<r>.

ConfigRule<Rule r>
=
( SpaceRule<r>      (. r.Type = RuleType.SpaceRule;      .)
| BreakRule<r>     (. r.Type = RuleType.BreakRule;     .)
| OpenRule<r>      (. r.Type = RuleType.OpenRule;      .)
| CloseRule<r>     (. r.Type = RuleType.CloseRule;     .)
| HideRule<r>      (. r.Type = RuleType.HideRule;      .)
| OverlapRule<r>   (. r.Type = RuleType.OverlapRule;   .)
)
.

HyperConfigRule<Rule r>
=
( StyleRule<r>      (. r.Type = RuleType.StyleRule;      .)
| ClassRule<r>     (. r.Type = RuleType.ClassRule;     .)
)
.

SpaceRule<Rule r>
= "space"
  SymbolAllIndent<out r.FirstSymbol>
  SymbolAll<out r.SecondSymbol>
  number              (. r.Number = Convert.ToInt32(t.val); .)
.

BreakRule<Rule r>
= "break"
  SymbolAll<out r.FirstSymbol>
  SymbolAll<out r.SecondSymbol>
  ( number            (. r.Number = Convert.ToInt32(t.val); .)
  | "default"         (. r.BreakMode = BreakMode.Default; .)
  | "always"          (. r.BreakMode = BreakMode.Always; .)
  | "never"           (. r.BreakMode = BreakMode.Never; .)
  )
.

OpenRule<Rule r>
= "open"
  (                  (. r.CreateBitSet();          .)
  | BitSet<r>
  | Bit<out r.FirstSymbol>
  )
.

CloseRule<Rule r>
= "close"
  (                  (. r.CreateBitSet();          .)
  | BitSet<r>
  | Bit<out r.FirstSymbol>
  )

```



```

.
HideRule<Rule r>
= "hide"
(
    BitSet<r> (. r.CreateBitSet(); .)
    |
    Bit<out r.FirstSymbol>
)
.
OverlapRule<Rule r>
= "overlap"
(
    BitSet<r> (. r.CreateBitSet(); .)
    |
    Bit<out r.FirstSymbol>
)
.
StyleRule<Rule r>
= "style"
(
    BitSet<r> (. r.CreateBitSet(); .)
    |
    Bit<out r.FirstSymbol>
)
.
ClassRule<Rule r>
= "class"
Text (. r.Text = t.val; .)
[
    SymbolAll<out r.FirstSymbol>
]
Text (. r.FurtherText = new ArrayList();
r.FurtherText.Add(t.val); .)

{
    Text (. r.FurtherText.Add(t.val); .)
}
.
Bit<out Symbol sym>
= SymbolAll <out sym>
.
BitSet<Rule rule>
= (. Symbol sym = Symbol.UNSET; .)
['
{ SingleSymbol<out sym> (. rule.Add(sym); .)
}
']
.
SymbolAll<out Symbol sym>
= (. sym = Symbol.UNSET; .)
( "ALL" (. sym = Symbol.ALL; .)
| SingleSymbol<out sym>
)
.
SymbolAllIndent<out Symbol sym>
= (. sym = Symbol.UNSET; .)
( "INDENT" (. sym = Symbol.INDENT; .)
| SymbolAll<out sym>
)
.

```

```


SingleSymbol<out Symbol sym>
=
(
  "Production"      (. sym = Symbol.UNSET; .)
  "Symbol"          (. sym = Symbol.Production; .)
  "Literal"         (. sym = Symbol.Symbol; .)
  "Open"            (. sym = Symbol.Literal; .)
  "Close"           (. sym = Symbol.Open; .)
  "Assign"          (. sym = Symbol.Close; .)
  "Separator"       (. sym = Symbol.Assign; .)
  "Delimiter"       (. sym = Symbol.Separator; .)
  "InnerComment"   (. sym = Symbol.Delimiter; .)
  "OuterComment"   (. sym = Symbol.InnerComment; .)
  "Attribute"       (. sym = Symbol.OuterComment; .)
  "Resolver"        (. sym = Symbol.Attribute; .)
  "Prolog"          (. sym = Symbol.Resolver; .)
  "Epilog"          (. sym = Symbol.Prolog; .)
  "Action"          (. sym = Symbol.Epilog; .)
  "LineNumber"     (. sym = Symbol.Action; .)
  "LineNumber"     (. sym = Symbol.LineNumber; .)
)

Text
= '<'
  { ANY }
  '>'

END PrintyConfig.






```

Constructors


Constructor	Description
 Parser (see page 20)	Erzeugt ein neues Parser-Objekt

Members



Fields

Field	Description
 errors (see page 20)	Eine Instanz für ein Errors (see page 13) -Objekt, in welchem alle aufgetretenen Fehler verwaltet werden.
 la (see page 20)	lookahead token
 Rules (see page 20)	Eine Collection vom Typ ArrayList, in der alle Rule (see page 24) -Objekte gespeichert werden, die während des Parsens erzeugt worden sind.
 scanner (see page 20)	Der Scanner (see page 33) , der zum Lesen der Datei verwendet wird und entsprechende Tokens liefert.
 t (see page 20)	last recognized token



Constructors

Constructor	Description
 Parser (see page 20)	Erzeugt ein neues Parser-Objekt

Methods

Method	Description
 Parse (see page 21)	Mit dieser Methode wird der Parse-Vorgang initialisiert.
 SemErr (see page 21)	Erzeugt einen neuen Semantik-Error

Legend

	Data Member
	Method

1.1.8.1 Fields

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

1.1.8.1.1 Parser.errors Field

```
public Errors errors;
```

Remarks

Eine Instanz für ein Errors (see page 13)-Objekt, in welchem alle aufgetretenen Fehler verwaltet werden.

1.1.8.1.2 Parser.la Field

```
public Token la;
```

Remarks

lookahead token

1.1.8.1.3 Parser.Rules Field

```
public ArrayList Rules;
```

Remarks

Eine Collection vom Typ ArrayList, in der alle Rule (see page 24)-Objekte gespeichert werden, die während des Parsens erzeugt worden sind.

1.1.8.1.4 Parser.scanner Field

```
public Scanner scanner;
```

Remarks

Der Scanner (see page 33), der zum Lesen der Datei verwendet wird und entsprechende Tokens liefert.

1.1.8.1.5 Parser.t Field

```
public Token t;
```

Remarks

last recognized token

1.1.8.2 Constructors

1.1.8.2.1 Parser.Parser Constructor

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

Remarks

Erzeugt ein neues Parser-Objekt

1.1.8.3 Methods

1.1.8.3.1 Parser.Parse Method

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

Remarks

Mit dieser Methode wird der Parse-Vorgang initialisiert.

1.1.8.3.2 Parser.SemErr Method

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

Remarks

Erzeugt einen neuen Semantik-Error

1.1.9 ParserError Class

Class Hierarchy

```
net.stefanhauser.Printy.GUI.Config.Parse...
```

```
public class ParserError;
```


File

Parser.cs (see page 51)

Remarks


Ein Objekt dieser Klasse wird erzeugt, wenn im Parser (see page 16) ein Fehler festgestellt wird.

Constructors

Constructor	Description
 <code>ParserError</code> (see page 22)	Erzeugt ein neues ParserError-Objekt

Members





Constructors

Constructor	Description
 <code>ParserError</code> (see page 22)	Erzeugt ein neues ParserError-Objekt



Methods

Method	Description
 <code>ToString</code> (see page 22)	Liefert eine String-Repräsentation des Fehlers

Properties

Property	Description
 <code>Column</code> (see page 22)	Die Spalte, in welcher der Fehler aufgetreten ist
 <code>Line</code> (see page 22)	Die Zeile, in welcher der Fehler aufgetreten ist
 <code>Text</code> (see page 22)	Eine textuelle Beschreibung des Fehlers
 <code>Type</code> (see page 22)	Liefert den Typ des entstandenen Fehlers

Legend

	Method
	Property

1.1.9.1 Constructors

1.1.9.1.1 ParserError.ParserError Constructor

```
public ParserError(int line, int col, string text, ErrorType type);
```

Remarks

Erzeugt ein neues ParserError-Objekt

1.1.9.2 Methods

1.1.9.2.1 ParserError.ToString Method

```
public override string ToString();
```

Remarks

Liefert eine String-Repräsentation des Fehlers

1.1.9.3 Properties

1.1.9.3.1 ParserError.Column Property

```
public int Column;
```

Remarks

Die Spalte, in welcher der Fehler aufgetreten ist

1.1.9.3.2 ParserError.Line Property

```
public int Line;
```

Remarks

Die Zeile, in welcher der Fehler aufgetreten ist

1.1.9.3.3 ParserError.Text Property

```
public string Text;
```

Remarks

Eine textuelle Beschreibung des Fehlers

1.1.9.3.4 ParserError.Type Property

```
public ErrorType Type;
```

Remarks

Liefert den Typ des entstandenen Fehlers

1.1.10 ParserException Class

Class Hierarchy



```
public class ParserException : System.Exception;
```

File

Parser.cs (see page 51)

Remarks

Eine solche Exception wird geworfen, wenn während des Parse-Vorgangs mindestens ein Fehler gefunden worden ist. Diese Exception wird von der Methode Parser.Parse (see page 21)() geworfen.

Constructors

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

Members

Constructors

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

Properties

Property	Description
ErrorCount (see page 23)	Die Anzahl der aufgetretenen Fehler
ErrorList (see page 24)	Eine Liste vom Typ ArrayList mit allen aufgetretenen Fehlern während des Parsens.

Legend

	Method
	Property

1.1.10.1 Constructors

1.1.10.1.1 ParserException.ParserException Constructor

```
public ParserException(ArrayList list, int count);
```

Remarks

Erzeugt ein neues ParserException-Objekt

1.1.10.2 Properties

1.1.10.2.1 ParserException.ErrorCount Property

```
public int ErrorCount;
```

Remarks

Die Anzahl der aufgetretenen Fehler

1.1.10.2.2 ParseException.ErrorList Property

```
public ICollection ErrorList;
```

Remarks

Eine Liste vom Typ ArrayList mit allen aufgetretenen Fehlern während des Parsens.

1.1.11 Rule Class**Class Hierarchy**

```
net.stefanhauser.Printy.GUI.Config.Rule
```

```
public class Rule;
```

File

Rule.cs (🔗 see page 52)

Remarks

Ein Objekt dieser Klasse stellt eine Regel dar, wie sie über graphische Steuerelemente erstellt werden kann. Eine solche Regel hat einen Regeltyp und verschiedene Parameter. Sollte einer dieser Parameter



Members**Fields**

Field	Description
◆ BreakMode (🔗 see page 25)	Dieses Feld stellt die Break-Option dar und wird daher nur bei einer Break-Regel verwendet.
◆ FirstSymbol (🔗 see page 25)	Das zweite Symbol (🔗 see page 47) einer Regel. Dieses Feld wird bei Space- und Break-Regeln verwendet
◆ FurtherText (🔗 see page 25)	Eine Collection vom Typ ArrayList, in der ausschließlich String-Objekte gespeichert werden. Dieses Feld wird nur bei einer Class-Regel benutzt.
◆ Number (🔗 see page 25)	Der Integer-Wert, der bei einer Space-Regel oder einer Break-Regel zum Einsatz kommen kann.
◆ SecondSymbol (🔗 see page 25)	Das zweite Symbol (🔗 see page 47) einer Regel. Dieses Feld wird bei Space- und Break-Regeln verwendet
◆ Text (🔗 see page 25)	Der erste Text-Teil einer Class-Regel
◆ Type (🔗 see page 26)	Der Regeltyp

Methods

Method	Description
⇒◆ Add (🔗 see page 26)	Diese Methode fügt ein neues Symbol (🔗 see page 47) in das BitSet ein.
⇒◆ CreateBitSet (🔗 see page 26)	Erstellt ein neues BitSet - Feld vom Typ ArrayList. Diese Funktion wird exklusiv vom Parser (🔗 see page 16) aufgerufen.
⇒◆ Equals (🔗 see page 26)	Prüft, ob zwei Regeln gleich sind. Dieser Vergleich beruht auf den Vergleich von deren String-Repräsentation
⇒◆ IsValid (🔗 see page 27)	Diese Methode überprüft, ob eine gültige Regel erstellt worden ist. Zur Überprüfung werden die Regeln aus der ATG-Datei verwendet
⇒◆ ToString (🔗 see page 27)	Liefert eine String-Repräsentation dieser Regel. Diese Repräsentation wird auch für die Serialisierung in die Regel - Datei verwendet.

Legend

	Data Member
	Method

1.1.11.1 Fields

1.1.11.1.1 Rule.BreakMode Field

```
public BreakMode BreakMode = BreakMode.UNSET;
```

Remarks

Dieses Feld stellt die Break-Option dar und wird daher nur bei einer Break-Regel verwendet.

1.1.11.1.2 Rule.FirstSymbol Field

```
public Symbol FirstSymbol = Symbol.UNSET;
```

Remarks

Das zweite Symbol (☞ see page 47) einer Regel. Dieses Feld wird bei Space- und Break-Regeln verwendet

1.1.11.1.3 Rule.FurtherText Field

```
public ArrayList FurtherText;
```

Remarks

Eine Collection vom Typ ArrayList, in der ausschließlich String-Objekte gespeichert werden. Dieses Feld wird nur bei einer Class-Regel benutzt.

1.1.11.1.4 Rule.Number Field

```
public int Number = -1;
```

Remarks

Der Integer-Wert, der bei einer Space-Regel oder einer Break-Regel zum Einsatz kommen kann.

1.1.11.1.5 Rule.SecondSymbol Field

```
public Symbol SecondSymbol = Symbol.UNSET;
```

Remarks

Das zweite Symbol (☞ see page 47) einer Regel. Dieses Feld wird bei Space- und Break-Regeln verwendet

1.1.11.1.6 Rule.Text Field

```
public string Text = String.Empty;
```

Remarks

Der erste Text-Teil einer Class-Regel

1.1.11.1.7 Rule.Type Field

```
public RuleType Type = RuleType.UNSET;
```

Remarks

Der Regeltyp

1.1.11.2 Methods

1.1.11.2.1 Add

1.1.11.2.1.1 Rule.Add Method (Symbol)

```
public void Add(Symbol s);
```

Remarks

Diese Methode fügt ein neues Symbol (☞ see page 47) in das BitSet ein.

1.1.11.2.1.2 Rule.Add Method (Wrapper)

```
public void Add(Wrapper w);
```

Remarks

Fügt ein neues Wrapper (☞ see page 42)-Objekt in die enthaltene Collection ein.

1.1.11.2.2 Rule.CreateBitSet Method

```
public void CreateBitSet();
```

Remarks

Erstellt ein neues BitSet - Feld vom Typ ArrayList. Diese Funktion wird exklusiv vom Parser (☞ see page 16) aufgerufen.

1.1.11.2.3 Equals

1.1.11.2.3.1 Rule.Equals Method (Rule)

Prüft, ob zwei Regeln gleich sind. Dieser Vergleich beruht auf den Vergleich von deren String-Repräsentation

```
public bool Equals(Rule other);
```

Parameters

Parameters	Description
obj	Das Rule (☞ see page 24)-Objekt

1.1.11.2.3.2 Rule.Equals Method (object)

Prüft, ob zwei Regeln gleich sind. Dieser Vergleich beruht auf den Vergleich von deren String-Repräsentation

```
public override bool Equals(object obj);
```

Parameters

Parameters	Description
object obj	Das Rule (see page 24)-Objekt

1.1.11.2.4 Rule.IsValid Method

```
public bool IsValid();
```

Remarks

Diese Methode überprüft, ob eine gültige Regel erstellt worden ist. Zur Überprüfung werden die Regeln aus der ATG-Datei verwendet

1.1.11.2.5 Rule.ToString Method

```
public override string ToString();
```

Remarks

Liefert eine String-Repräsentation dieser Regel. Diese Repräsentation wird auch für die Serialisierung in die Regel - Datei verwendet.

1.1.12 RuleControl Class

Dieses Steuerelement ermöglicht dem Benutzer, Regeln aller Art zu erstellen. Es enthält eine ComboBox, welche alle Regeltypen auflistet. Sobald ein Regeltyp gewählt worden ist, wird das entsprechende Steuerelement angezeigt und in einer entsprechenden Variablen-Instanz gespeichert.

Class Hierarchy

```
public class RuleControl : System.Windows.Forms.UserControl, IHelpful;
```

File

RuleControl.cs (see page 52)

Constructors

Constructor	Description
RuleControl (see page 28)	Erzeugt ein neues RuleControl-Steuerelement

Members**Fields**

Field	Description
ActiveControl (see page 28)	Das momentan aktive Steuerelement, mit dem eine neue Regel erstellt werden soll

Constructors

Constructor	Description
RuleControl (see page 28)	Erzeugt ein neues RuleControl-Steuerelement


Methods

Method	Description
Dispose (see page 28)	Clean up any resources being used.





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>.

 SetupHelpProvider (see page 28)	This is SetupHelpProvider, a member of class RuleControl.
---	---

Properties

Property	Description
 Enabled (see page 29)	This is Enabled, a member of class RuleControl.

Legend

	Data Member
	Method
	protected
	Property

1.1.12.1 Fields

1.1.12.1.1 RuleControl.ActiveControl Field

```
public new AbstractRuleControl ActiveControl;
```

Remarks

Das momentan aktive Steuerelement, mit dem eine neue Regel erstellt werden soll

1.1.12.2 Constructors

1.1.12.2.1 RuleControl.RuleControl Constructor

```
public RuleControl();
```

Remarks

Erzeugt ein neues RuleControl-Steuerelement

1.1.12.3 Methods

1.1.12.3.1 RuleControl.Dispose Method

Clean up any resources being used.

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

1.1.12.3.2 RuleControl.SetupHelpProvider Method

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

Remarks

This is SetupHelpProvider, a member of class RuleControl.

1.1.12.4 Properties

1.1.12.4.1 RuleControl.Enabled Property

```
public new bool Enabled;
```

Remarks

This is Enabled, a member of class RuleControl.

1.1.13 RuleControlFactory Class

Class Hierarchy

```
net.stefanhauser.Printy.GUI.Config.RuleC...
```

```
public class RuleControlFactory;
```

File


RuleControlFactory.cs (see page 52)

Remarks



This is class net.stefanhauser.Printy.GUI.Config.RuleControlFactory.

Members

Methods

Method	Description
  CreateRuleControl (see page 29)	This is CreateRuleControl, a member of class RuleControlFactory.

Legend

	Method
	static

1.1.13.1 Methods

1.1.13.1.1 RuleControlFactory.CreateRuleControl Method

```
public static AbstractRuleControl CreateRuleControl(RuleType type);
```

Remarks

This is CreateRuleControl, a member of class RuleControlFactory.

1.1.14 RuleEventArgs Class

Ein Objekt dieser Klasse wird erzeugt, wenn im RuleManager (see page 30) entweder eine neue Regel hinzugefügt oder entfernt worden ist.

Class Hierarchy


```
EventArgs → net.stefanhauser.Printy.GUI.Config.RuleE...
```

```
public class RuleEventArgs : EventArgs;
```


File

RuleManager.cs (see page 53)


Constructors

Constructor	Description
 RuleEventArgs (see page 30)	Erzeugt ein neues RuleEventArgs-Objekt



Members**Constructors**

Constructor	Description
 RuleEventArgs (see page 30)	Erzeugt ein neues RuleEventArgs-Objekt

Properties

Property	Description
 Rule (see page 30)	Die Regel, die hinzugefügt oder entfernt worden ist.

Legend

	Method
	Property

1.1.14.1 Constructors

1.1.14.1.1 RuleEventArgs.RuleEventArgs Constructor

```
public RuleEventArgs(Rule r);
```

Remarks

Erzeugt ein neues RuleEventArgs-Objekt

1.1.14.2 Properties

1.1.14.2.1 RuleEventArgs.Rule Property

```
public Rule Rule;
```

Remarks

Die Regel, die hinzugefügt oder entfernt worden ist.

1.1.15 RuleManager Class

Class Hierarchy

```
net.stefanhauser.Printy.GUI.Config.Rule...
```

```
public class RuleManager;
```

File

RuleManager.cs (see page 53)

Remarks

Diese Klasse verwaltet die Regeln, die über die graphische Benutzeroberfläche erstellt worden sind. Zur Speicherung der Rule (☞ see page 24)-Objekte wird eine Collection vom Typ ArrayList verwendet.

Weiters gibt es in dieser Klasse noch die Möglichkeit, den Namen der Datei zu spezifizieren, aus welcher die Regeln gelesen werden bz. in welche die Regeln geschrieben werden.

Constructors

Constructor	Description
☞ RuleManager (☞ see page 31)	Erzeugt ein neues RuleManager-Objekt.

Members

Constructors

Constructor	Description
☞ RuleManager (☞ see page 31)	Erzeugt ein neues RuleManager-Objekt.

Methods

Method	Description
☞ Add (☞ see page 32)	Fügt eine neue Regel in die Collection ein. Dabei werden alle registrierten Listener alarmiert.
☞ Remove (☞ see page 32)	Entfernt eine Regel aus der Collection. Dabei werden alle registrierten Listener aktiviert.
☞ RemoveAll (☞ see page 32)	This is RemoveAll, a member of class RuleManager.
☞ WriteToFile (☞ see page 32)	This is WriteToFile, a member of class RuleManager.

Properties

Property	Description
☞ Filename (☞ see page 32)	Setzt oder liefert den Namen der Datei, die als Quelle und als Speicherziel für die Regeln verwendet wird. Sobald eine neue Datei spezifiziert wird, werden mit Hilfe des Parsers alle Regeln aus der Datei gelesen.
☞ TempFileName (☞ see page 33)	This is TempFileName, a member of class RuleManager.

Events

Event	Description
⚡ FileNameChanged (☞ see page 33)	Dieser Event wird ausgelöst, wenn der Name der Quelldatei geändert worden ist.
⚡ RuleAdded (☞ see page 33)	Dieser Event wird aufgerufen, wenn eine Regel zur Colection hinzugefügt worden ist.
⚡ RuleRemoved (☞ see page 33)	Dieser Event wird ausgelöst, wenn eine Regel aus dem Manager entfernt worden ist.

Legend

☞	Method
☞	Property
⚡	Event

1.1.15.1 Constructors

1.1.15.1.1 RuleManager.RuleManager Constructor

```
public RuleManager();
```

Remarks

Erzeugt ein neues RuleManager-Objekt.

1.1.15.2 Methods

1.1.15.2.1 RuleManager.Add Method

```
public void Add(Rule r);
```

Remarks

Fügt eine neue Regel in die Collection ein. Dabei werden alle registrierten Listener alarmiert.

1.1.15.2.2 RuleManager.Remove Method

```
public void Remove(Rule r);
```

Remarks

Entfernt eine Regel aus der Collection. Dabei werden alle registrierten Listener aktiviert.

1.1.15.2.3 RuleManager.RemoveAll Method

```
public void RemoveAll();
```

Remarks

This is RemoveAll, a member of class RuleManager.

1.1.15.2.4 WriteToFile

1.1.15.2.4.1 RuleManager.WriteToFile Method ()

```
public void WriteToFile();
```

Remarks

This is WriteToFile, a member of class RuleManager.

1.1.15.2.4.2 RuleManager.WriteToFile Method (string)

```
public void WriteToFile(string name);
```

Remarks

Schreibt alle Regeln, die in der Collection gespeichert sind, in eine Datei. Dabei wird der Dateiname verwendet, welcher über das entsprechende Property festgelegt wurde.

1.1.15.3 Properties

1.1.15.3.1 RuleManager.FileName Property

```
public string FileName;
```

Remarks

Setzt oder liefert den Namen der Datei, die als Quelle und als Speicherziel für die Regeln verwendet wird. Sobald eine neue Datei spezifiziert wird, werden mit Hilfe des Parsers alle Regeln aus der Datei gelesen.

1.1.15.3.2 RuleManager.TempFileName Property

```
public string TempFileName;
```

Remarks

This is TempFileName, a member of class RuleManager.

1.1.15.4 Events

1.1.15.4.1 RuleManager.FileNameChanged Event

```
public event FilenameEventHandler FileNameChanged;
```

Remarks

Dieser Event wird ausgelöst, wenn der Name der Quelldatei geändert worden ist.

1.1.15.4.2 RuleManager.RuleAdded Event

```
public event RuleEventHandler RuleAdded;
```

Remarks

Dieser Event wird aufgerufen, wenn eine Regel zur Colection hinzugefügt worden ist.

1.1.15.4.3 RuleManager.RuleRemoved Event

```
public event RuleEventHandler RuleRemoved;
```

Remarks

Dieser Event wird ausgelöst, wenn eine Regel aus dem Manager entfernt worden ist.

1.1.16 Scanner Class

Class Hierarchy

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

```
public class Scanner;
```

File

Scanner.cs (📄 see page 53)

Remarks

Der Scanner, der zum Lesen der Regel-Datei verwendet wird. Diese Klasse wurde durch das Tool COCO erstellt.

```
/*-----
Config.atg
2005-11 Stefan Hauser

Original Version by Philip Grasböck
-----*/
```



```

COMPILER PrintyConfig

CHARACTERS
  digit      = "0123456789".
  tab        = '\t'.
  cr         = '\r'.
  lf         = '\n'.
  eol        = cr + lf.

TOKENS
  number = ['-'] digit { digit }.

COMMENTS FROM "/*" TO "*/" NESTED
COMMENTS FROM "//" TO cr lf

IGNORE eol + tab

/*-----*/

PRODUCTIONS

PrintyConfig
=
{
    Rule<r>
    ';'
}

Rule<Rule r> = ConfigRule<r> | HyperConfigRule<r>.

ConfigRule<Rule r>
=
(
  SpaceRule<r>      (. r.Type = RuleType.SpaceRule;      .)
  BreakRule<r>     (. r.Type = RuleType.BreakRule;      .)
  OpenRule<r>      (. r.Type = RuleType.OpenRule;       .)
  CloseRule<r>     (. r.Type = RuleType.CloseRule;      .)
  HideRule<r>      (. r.Type = RuleType.HideRule;       .)
  OverlapRule<r>   (. r.Type = RuleType.OverlapRule;    .)
)

HyperConfigRule<Rule r>
=
(
  StyleRule<r>     (. r.Type = RuleType.StyleRule;      .)
  ClassRule<r>    (. r.Type = RuleType.ClassRule;       .)
)

SpaceRule<Rule r>
= "space"
  SymbolAllIndent<out r.FirstSymbol>
  SymbolAll<out r.SecondSymbol>
  number (. r.Number = Convert.ToInt32(t.val); .)

BreakRule<Rule r>
= "break"
  SymbolAll<out r.FirstSymbol>
  SymbolAll<out r.SecondSymbol>
  (
    number (. r.Number = Convert.ToInt32(t.val); .)
    "default" (. r.BreakMode = BreakMode.Default; .)
    "always" (. r.BreakMode = BreakMode.Always; .)
    "never" (. r.BreakMode = BreakMode.Never; .)
  )

```

```

OpenRule<Rule r>
= "open"
(
    (. r.CreateBitSet();
    BitSet<r>
    |
    Bit<out r.FirstSymbol>
    )
.

CloseRule<Rule r>
= "close"
(
    (. r.CreateBitSet();
    BitSet<r>
    |
    Bit<out r.FirstSymbol>
    )
.

HideRule<Rule r>
= "hide"
(
    (. r.CreateBitSet();
    BitSet<r>
    |
    Bit<out r.FirstSymbol>
    )
.

OverlapRule<Rule r>
= "overlap"
(
    (. r.CreateBitSet();
    BitSet<r>
    |
    Bit<out r.FirstSymbol>
    )
.

StyleRule<Rule r>
= "style"
(
    (. r.CreateBitSet();
    BitSet<r>
    |
    Bit<out r.FirstSymbol>
    )
.

ClassRule<Rule r>
= "class"
Text
    (. r.Text = t.val;
    [
    SymbolAll<out r.FirstSymbol>
    ]
Text
    (. r.FurtherText = new ArrayList();
    r.FurtherText.Add(t.val);
    {
    Text
    (. r.FurtherText.Add(t.val);
    }
.

Bit<out Symbol sym>
= SymbolAll <out sym>
.

BitSet<Rule rule>
=
    (. Symbol sym = Symbol.UNSET;
    '['
    { SingleSymbol<out sym>
    (. rule.Add(sym);

```

```

    }
    ']'
    .

SymbolAll<out Symbol sym>
=
    ( "ALL" (. sym = Symbol.UNSET; .)
      | SingleSymbol<out sym>
    )
    .

SymbolAllIndent<out Symbol sym>
=
    ( "INDENT" (. sym = Symbol.UNSET; .)
      | SymbolAll<out sym>
    )
    .

SingleSymbol<out Symbol sym>
=
    ( "Production" (. sym = Symbol.UNSET; .)
      | "Symbol" (. sym = Symbol.Symbol; .)
      | "Literal" (. sym = Symbol.Literal; .)
      | "Open" (. sym = Symbol.Open; .)
      | "Close" (. sym = Symbol.Close; .)
      | "Assign" (. sym = Symbol.Assign; .)
      | "Separator" (. sym = Symbol.Separator; .)
      | "Delimiter" (. sym = Symbol.Delimiter; .)
      | "InnerComment" (. sym = Symbol.InnerComment; .)
      | "OuterComment" (. sym = Symbol.OuterComment; .)
      | "Attribute" (. sym = Symbol.Attribute; .)
      | "Resolver" (. sym = Symbol.Resolver; .)
      | "Prolog" (. sym = Symbol.Prolog; .)
      | "Epilog" (. sym = Symbol.Epilog; .)
      | "Action" (. sym = Symbol.Action; .)
      | "LineNumber" (. sym = Symbol.LineNumber; .)
    )
    .

Text
= '<'
  { ANY }
  '>'
    .

END PrintyConfig.

```

Constructors

Constructor	Description
Scanner (see page 37)	Erzeugt ein neues Scanner-Objekt

Members

Fields

Field	Description
buffer (see page 37)	scanner buffer

Constructors



Constructor	Description
Scanner (see page 37)	Erzeugt ein neues Scanner-Objekt

Methods

Method	Description
Peek (see page 37)	peek for the next token, ignore pragmas
ResetPeek (see page 37)	make sure that peeking starts at the current scan position

 Scan (see page 38)	get the next token (possibly a token already seen during peeking)
--	---

Legend

	Data Member
	Method

1.1.16.1 Fields

1.1.16.1.1 Scanner.buffer Field

```
public Buffer buffer;
```

Remarks

scanner buffer

1.1.16.2 Constructors

1.1.16.2.1 Scanner

1.1.16.2.1.1 Scanner.Scanner Constructor (Stream)

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

Remarks

Erzeugt ein neues Scanner-Objekt

1.1.16.2.1.2 Scanner.Scanner Constructor (string)

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

Remarks

Erzeugt ein neues Scanner-Objekt

1.1.16.3 Methods

1.1.16.3.1 Scanner.Peek Method

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

Remarks

peek for the next token, ignore pragmas

1.1.16.3.2 Scanner.ResetPeek Method

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

Remarks

make sure that peeking starts at the current scan position

1.1.16.3.3 Scanner.Scan Method

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

Remarks

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

1.1.17 SpaceRuleControl Class

Mit diesem Steuerelement ist es möglich, neue Regeln vom Typ SpaceRule zu erstellen

Class Hierarchy



```
public class SpaceRuleControl : AbstractRuleControl;
```

File

SpaceRuleControl.cs (see page 54)

Constructors

Constructor	Description
SpaceRuleControl (see page 39)	Erzeugt ein neues SpaceRule-Steuerelement

Members

Constructors

Constructor	Description
SpaceRuleControl (see page 39)	Erzeugt ein neues SpaceRule-Steuerelement

Methods

Method	Description
Dispose (see page 39)	Clean up any resources being used.
GetCreatedRule (see page 39)	Liefert das Rule (see page 24)-Objekt, welches mit dem Steuerelement erzeugt worden ist

AbstractRuleControl Class

AbstractRuleControl Class	Description
GetCreatedRule (see page 4)	Diese Methode liefert das Rule (see page 24)-Objekt, welches mit Hilfe der verschiedenen Steuerelemente erstellt worden ist. Diese Methode ist als abstrakt deklariert und muss in jeder erbenden Klasse implementiert werden.
SetupHelpProvider (see page 4)	This is SetupHelpProvider, a member of class AbstractRuleControl.

IRuleControl Interface

IRuleControl Interface	Description
GetCreatedRule (see page 45)	Liefert die erstellte Regel

Fields

AbstractRuleControl Class

AbstractRuleControl Class	Description
groupBox1 (see page 3)	Die GroupBox, welche alle Steuerelemente enthält, mit deren Hilfe man Regeln erstellen kann
ruleType (see page 4)	Der Typ der zu erstellenden Regel


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

Properties






AbstractRuleControl Class

AbstractRuleControl Class	Description
 Text  see page 4)	Setzt oder liefert den Namen der Regel, die im Steuerelement erstellt werden kann

IRuleControl Interface

IRuleControl Interface	Description
 Text  see page 46)	Setzt oder liefert den Typ der Regel

Legend

	Method
	protected
	abstract
	Data Member
	Property

1.1.17.1 Constructors

1.1.17.1.1 SpaceRuleControl.SpaceRuleControl Constructor

```
public SpaceRuleControl();
```

Remarks

Erzeugt ein neues SpaceRule-Steuerelement

1.1.17.2 Methods

1.1.17.2.1 SpaceRuleControl.Dispose Method


Clean up any resources being used.

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

1.1.17.2.2 SpaceRuleControl.GetCreatedRule Method

```
public override Rule GetCreatedRule();
```

Remarks

Liefert das Rule ( see page 24)-Objekt, welches mit dem Steuerelement erzeugt worden ist

1.1.18 Token Class

Class Hierarchy

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

```
public class Token;
```

File

Scanner.cs (see page 53)


Remarks

Token-Objekte werden vom Scanner (see page 33) während des Lesevorgangs geliefert

Members**Fields**

Field	Description
col (see page 40)	token column (starting at 0)
kind (see page 40)	token kind
line (see page 40)	token line (starting at 1)
next (see page 40)	ML 2005-03-11 Tokens are kept in linked list
pos (see page 41)	token position in the source text (starting at 0)
val (see page 41)	token value

Legend

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

1.1.18.1 Fields

1.1.18.1.1 Token.col Field

```
public int col;
```

Remarks

token column (starting at 0)

1.1.18.1.2 Token.kind Field

```
public int kind;
```

Remarks

token kind

1.1.18.1.3 Token.line Field

```
public int line;
```

Remarks

token line (starting at 1)

1.1.18.1.4 Token.next Field

```
public Token next;
```

Remarks

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

1.1.18.1.5 Token.pos Field

```
public int pos;
```

Remarks

token position in the source text (starting at 0)

1.1.18.1.6 Token.val Field

```
public string val;
```

Remarks

token value

1.1.19 Utilities Class

Diese Klasse enthält verschiedene Sammlungen von Enumationswerten, wie sie im Parser (☞ see page 16) verwendet und gelesen werden.

Class Hierarchy

```
net.stefanhauser.Printy.GUI.Config.Utilities
```









```
public class Utilities;
```

File



Utilities.cs (☞ see page 54)

Members

Properties

Property	Description
  BreakRules (☞ see page 41)	Die Menge BreakRules mit Elementen vom Typ BreakRule
  Rules (☞ see page 42)	Die Menge Rules mit Elementen vom Typ RuleType (☞ see page 47)
  SingleSymbol (☞ see page 42)	Die Menge SingleSymbol mit Elementen vom Typ Symbol (☞ see page 47)
  SymbolAll (☞ see page 42)	Die Menge SymbolAll mit Elementen vom Typ Symbol (☞ see page 47)
  SymbolAllIndent (☞ see page 42)	Die Menge SymbolAllIndent mit Elementen vom Typ Symbol (☞ see page 47)

Legend

	Property
	static

1.1.19.1 Properties

1.1.19.1.1 Utilities.BreakRules Property

```
public static Wrapper BreakRules;
```

Remarks

Die Menge BreakRules mit Elementen vom Typ BreakRule

1.1.19.1.2 Utilities.Rules Property

```
public static Wrapper Rules;
```

Remarks

Die Menge Rules mit Elementen vom Typ RuleType (☒ see page 47)

1.1.19.1.3 Utilities.SingleSymbol Property

```
public static Wrapper SingleSymbol;
```

Remarks

Die Menge SingleSymbol mit Elementen vom Typ Symbol (☒ see page 47)

1.1.19.1.4 Utilities.SymbolAll Property

```
public static Wrapper SymbolAll;
```

Remarks

Die Menge SymbolAll mit Elementen vom Typ Symbol (☒ see page 47)

1.1.19.1.5 Utilities.SymbolAllIndent Property

```
public static Wrapper SymbolAllIndent;
```

Remarks

Die Menge SymbolAllIndent mit Elementen vom Typ Symbol (☒ see page 47)

1.1.20 Wrapper Class

Class Hierarchy

```
net.stefanhauser.Printy.GUI.Config.Wrapper
```

```
public class Wrapper;
```

File

Wrapper.cs (☒ see page 54)

Remarks

Diese Klasse dient in erster Linie dazu, dass Enumerationen vom Typ RuleType (☒ see page 43), Symbol (☒ see page 43) und BreakMode (☒ see page 43) auch als Objekte verwendet werden können. Verschiedene Steuerelemente wie zum Beispiel eine ListBox akzeptieren für darzustellende Items lediglich object[]-Elemente.




Um die Verwendung dieser Klasse auch innerhalb des SourceCodes so weit wie möglich zu verstecken, gibt es in dieser Klasse zahlreiche Methoden zur impliziten und expliziten Typumwandlung.

Constructors


Constructor	Description
☒ Wrapper (☒ see page 44)	Erzeugt ein neues Wrapper-Objekt mit einem BreakMode (☒ see page 43)-Wert

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>.


Members**Fields**

Field	Description
 BreakMode (see page 43)	
 RuleType (see page 43)	
 Symbol (see page 43)	





Constructors

Constructor	Description
 Wrapper (see page 44)	Erzeugt ein neues Wrapper-Objekt mit einem BreakMode (see page 43)-Wert



Methods

Method	Description
 ToString (see page 44)	Liefert den Namen des "eingepackten" Enumerationwertes




Operators

Operator	Description
  BreakMode (see page 44)	Wandelt eine BreakMode-Variable in einen Wrapper um
  RuleType (see page 45)	Wandelt ein Wrapper-Objekt in eine RuleType-Variable um
  Symbol (see page 45)	Wandelt ein Wrapper-Objekt in eine Symbol-Variable um

Operators

Operator	Description
  BreakMode (see page 44)	Wandelt eine BreakMode-Variable in einen Wrapper um
  RuleType (see page 45)	Wandelt ein Wrapper-Objekt in eine RuleType-Variable um
  Symbol (see page 45)	Wandelt ein Wrapper-Objekt in eine Symbol-Variable um

Legend

	Data Member
	Method
	static

1.1.20.1 Fields

1.1.20.1.1 Wrapper.BreakMode Field

```
public BreakMode BreakMode = BreakMode.UNSET;
```

1.1.20.1.2 Wrapper.RuleType Field

```
public RuleType RuleType = RuleType.UNSET;
```

1.1.20.1.3 Wrapper.Symbol Field

```
public Symbol Symbol = Symbol.UNSET;
```

1.1.20.2 Constructors

1.1.20.2.1 Wrapper

1.1.20.2.1.1 Wrapper Wrapper Constructor (BreakMode)

```
public Wrapper(BreakMode br);
```

Remarks

Erzeugt ein neues Wrapper-Objekt mit einem BreakMode (☞ see page 43)-Wert

1.1.20.2.1.2 Wrapper Wrapper Operator (BreakMode)

Wandelt eine BreakMode (☞ see page 43)-Variable in einen Wrapper um

```
public static implicit operator Wrapper(BreakMode br);
```

1.1.20.2.1.3 Wrapper Wrapper Constructor (RuleType)

```
public Wrapper(RuleType rt);
```

Remarks

Erzeugt ein neues Wrapper-Objekt mit einem RuleType (☞ see page 43)-Wert

1.1.20.2.1.4 Wrapper Wrapper Operator (RuleType)

Wandelt eine RuleType (☞ see page 43)-Variable in einen Wrapper um

```
public static implicit operator Wrapper(RuleType r);
```

1.1.20.2.1.5 Wrapper Wrapper Constructor (Symbol)

```
public Wrapper(Symbol sym);
```

Remarks

Erzeugt ein neues Wrapper-Objekt mit einem Symbol (☞ see page 43)-Wert

1.1.20.2.1.6 Wrapper Wrapper Operator (Symbol)

Wandelt eine Symbol (☞ see page 43)-Variable in einen Wrapper um

```
public static implicit operator Wrapper(Symbol sym);
```

1.1.20.3 Methods

1.1.20.3.1 Wrapper.ToString Method

```
public override string ToString();
```

Remarks

Liefert den Namen des "eingepackten" Enumerationwertes

1.1.20.4 Operators

1.1.20.4.1 Wrapper.BreakMode Operator

Wandelt eine BreakMode-Variable in einen Wrapper (☞ see page 42) um

```
public static implicit operator BreakMode(Wrapper w);
```

1.1.20.4.2 Wrapper.RuleType Operator

Wandelt ein Wrapper (☞ see page 42)-Objekt in eine RuleType-Variable um

```
public static implicit operator RuleType(Wrapper w);
```

1.1.20.4.3 Wrapper.Symbol Operator

Wandelt ein Wrapper (☞ see page 42)-Objekt in eine Symbol-Variable um

```
public static implicit operator Symbol(Wrapper w);
```

1.1.21 IRuleControl Interface

Class Hierarchy

```
net.stefanhauser.Printy.GUI.Config.IRule...
```

```
public interface IRuleControl;
```

File

Enumerations.cs (☞ see page 51)

Remarks

Dieses Interface wird von jedem Steuerelement implementiert, welches in der Lage ist, eine Regel zu erstellen.

Members



Methods

Method	Description
 GetCreatedRule (☞ see page 45)	Liefert die erstellte Regel

Properties

Property	Description
 Text (☞ see page 46)	Setzt oder liefert den Typ der Regel

Legend

	Method
	Property

1.1.21.1 Methods

1.1.21.1.1 IRuleControl.GetCreatedRule Method

```
Rule GetCreatedRule();
```

Remarks

Liefert die erstellte Regel

1.1.21.2 Properties

1.1.21.2.1 IRuleControl.Text Property

```
RuleType Text;
```

Remarks

Setzt oder liefert den Typ der Regel

1.1.22 net.stefanhauser.Printy.GUI.Config.BreakMode Enumeration

```
public enum BreakMode {
    public Always,
    public Default,
    public Never,
    public UNSET
}
```

File

Enumerations.cs (see page 51)

Members

Members	Description
public Always	immer
public Default	default
public Never	nie
public UNSET	Der Standardwert dieser Enumeration

Remarks

Eine Enumeration, die alle Werte enthält, die eine BreakRule annehmen kann

1.1.23 net.stefanhauser.Printy.GUI.Config.ErrorType Enumeration

```
public enum ErrorType {
    public SemanticError,
    public SyntacticError,
    public Error
}
```

File

Parser.cs (see page 51)

Members

Members	Description
public SemanticError	Ein semantischer Fehler
public SyntacticError	Ein Syntax - Fehler
public Error	Ein allgemeiner Fehler

Remarks

Diese Enumeration enthält alle Fehlertypen, welche während des Parsens auftreten können.

1.1.24 net.stefanhauser.Printy.GUI.Config.RuleType Enumeration

```
public enum RuleType {
    public SpaceRule,
    public BreakRule,
    public OpenRule,
    public CloseRule,
    public HideRule,
    public OverlapRule,
    public StyleRule,
    public ClassRule,
    public UNSET
}
```

File

Enumerations.cs (see page 51)

Members

Members	Description
public SpaceRule	Abstands - Regel
public BreakRule	Umbruchsregel
public OpenRule	Öffnungsregel
public CloseRule	Schließungsregel
public HideRule	Regel, die beschreibt, welche Symbole nicht angezeigt werden sollen.
public OverlapRule	Überlappungs - Regel
public StyleRule	Stil - Regel für Stylesheet-Dokumente
public ClassRule	Klassenregel, falls HTML-Span-Klassen für die Formattierung verwendet werden sollen
public UNSET	Der Standardwert dieser Enumeration. Jedesmal, wenn eine Variable dieses Typs initialisiert wird, erhält sie diesen Wert.

Remarks

Diese Enumeration stellt eine Sammlung aller Typen von Regeln dar, mit welchen die Ausgabe des PrettyPrinters konfiguriert werden können.

1.1.25 net.stefanhauser.Printy.GUI.Config.Symbol Enumeration

```
public enum Symbol {
    public Production,
    public Symbol,
    public Literal,
    public Open,
    public Close,
    public Assign,
    public Seperator,
    public Delimiter,
    public InnerComment,
    public OuterComment,
    public Attribute,
    public Resolver,
    public Prolog,
    public Epilog,
}
```

```

    public Action,
    public LineNumber,
    public ALL,
    public INDENT,
    public UNSET
}

```

File

Enumerations.cs (🔗 see page 51)

Members

Members	Description
public Production	Production - Symbol
public Literal	Literal - Symbol
public Open	Open - Symbol
public Close	Close - Symbol
public Assign	Zuweisungs - Symbol
public Seperator	Seperator - Symbol
public Delimiter	Delimiter - Symbol
public InnerComment	InnerComment - Symbol
public OuterComment	OuterComment - Symbol
public Attribute	Attribut - Symbol
public Resolver	Resolver - Symbol
public Prolog	Prolog - Symbol
public Epilog	Epilog - Symbol
public Action	Action - Symbol
public LineNumber	Zeilennummer - Symbol
public ALL	Dieser Wert steht stellvertretend für alle Symbole dieser Enumeration
public INDENT	Einrückungs - Symbol
public UNSET	Der Standardwert dieser Enumeration. Er wird sofort nach der Instanzierung einer Variablen dieser Enumeration verwendet.

Remarks

Diese Enumeration enthält alle Symbole, welche in einer Konfigurationsdatei verwendet werden. Diese Symbole können zum Beispiel dafür verwendet werden, um eine Regel zu erstellen, mit der der Abstand zwischen zwei solchen Symbolen festgelegt werden kann.

1.1.26

net.stefanhauser.Printy.GUI.Config.RuleEventHandler Type

```
public delegate void RuleEventHandler(object sender, RuleEventArgs e);
```

File

RuleManager.cs (🔗 see page 53)

Remarks

Dieses Delegate wird verwendet, wenn eine Regel zum RuleManager (🔗 see page 30) hinzugefügt oder vom RuleManager (🔗 see page 30) entfernt worden ist.

1.2 Files

1.2.1 AbstractRuleControl.cs

This is file AbstractRuleControl.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (🔗 see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.2 AssemblyInfo.cs

This is file AssemblyInfo.cs.

1.2.3 BitSetControl.cs

This is file BitSetControl.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (🔗 see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.4 BreakRuleControl.cs

This is file BreakRuleControl.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.5 ClassControl.cs

This is file ClassControl.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.6 ConfigForm.cs

This is file ConfigForm.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.7 Enumerations.cs

This is file Enumerations.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.8 Parser.cs

This is file Parser.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.9 Rule.cs

This is file Rule.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.10 RuleControl.cs

This is file RuleControl.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.11 RuleControlFactory.cs

This is file RuleControlFactory.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.12 RuleManager.cs

This is file RuleManager.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.13 Scanner.cs

This is file Scanner.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.14 SpaceRuleControl.cs

This is file SpaceRuleControl.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.15 Utilities.cs

This is file Utilities.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

1.2.16 Wrapper.cs

This is file Wrapper.cs.

Namespaces

Namespace	Description
net.stefanhauser.Printy.GUI.Config (see page 1)	<p>Dieser Namespace enthält verschiedenste Elemente, um den Inhalt einer Datei zu erstellen, die zur Formatierung des PrettyPrinters dient. Der verwendete Parameter lautet</p> <pre>-config <filename></pre> <p>In diesem Namespace sind sowohl verschiedene Steuerelemente zum Erstellen diverser Regeln enthalten, als auch logische Elemente, um die erstellten Regeln zu verwalten.</p> <p>Dieser Namespace enthält gleichzeitig auch alle Elemente des Projektes. Das Projekt wird als Library kompiliert und vom Assembly PrintyUI verwendet.</p>

Index

A

AbstractRuleControl class 2

 GetCreatedRule 4

 groupBox1 3

 ruleType 4

 SetupHelpProvider 4

 Text 4

AbstractRuleControl.cs 49

AssemblyInfo.cs 49

B

BitSetControl class 4

 BitSetControl 6

 Dispose 6

 GetCreatedRule 6

BitSetControl.cs 49

BreakRuleControl class 6

 BreakRuleControl 7

 Dispose 7

 GetCreatedRule 8

BreakRuleControl.cs 50

Buffer class 8

 Buffer 9

 Close 9

 EOF 9

 GetString 9

 Peek 9

 Pos 9

 Read 9

C

ClassControl class 9

 ClassControl 11

 Dispose 11

 GetCreatedRule 11

ClassControl.cs 50

ConfigForm class 11

AdjustedParameter 13

ConfigFileName 13

ConfigForm 12

Dispose 12

Parameter 13

Reset 12

SetParameterEventHandler 12

SetupHelpProvider 13

ConfigForm.cs 50

E

Enumerations.cs 51

Errors class 13

 count 14

 CreateException 15

 errMsgFormat 14

 Error 15

 ErrorCount 16

 Errors 15

 Exception 15

 SemErr 15

 SynErr 16

I

IRuleControl interface 45

 GetCreatedRule 45

 Text 46

N

net.stefanhauser.Printy.GUI.Config 1

net.stefanhauser.Printy.GUI.Config namespace 1

net.stefanhauser.Printy.GUI.Config.AbstractRuleControl 2

net.stefanhauser.Printy.GUI.Config.AbstractRuleControl.GetCreatedRule 4

net.stefanhauser.Printy.GUI.Config.AbstractRuleControl.groupBox1 3

net.stefanhauser.Printy.GUI.Config.AbstractRuleControl.ruleType 4

net.stefanhauser.Printy.GUI.Config.AbstractRuleControl.Setup

HelpProvider 4	net.stefanhauser.Printy.GUI.Config.ConfigForm.ConfigForm 12
net.stefanhauser.Printy.GUI.Config.AbstractRuleControl.Text 4	net.stefanhauser.Printy.GUI.Config.ConfigForm.Dispose 12 net.stefanhauser.Printy.GUI.Config.ConfigForm.Parameter 13
net.stefanhauser.Printy.GUI.Config.BitSetControl 4	net.stefanhauser.Printy.GUI.Config.ConfigForm.Reset 12
net.stefanhauser.Printy.GUI.Config.BitSetControl.BitSetControl 6	net.stefanhauser.Printy.GUI.Config.ConfigForm.SetParameter EventHandler 12
net.stefanhauser.Printy.GUI.Config.BitSetControl.Dispose 6	net.stefanhauser.Printy.GUI.Config.ConfigForm.SetupHelpPro vider 13
net.stefanhauser.Printy.GUI.Config.BitSetControl.GetCreated Rule 6	net.stefanhauser.Printy.GUI.Config.Errors 13
net.stefanhauser.Printy.GUI.Config.BreakMode 46	net.stefanhauser.Printy.GUI.Config.Errors.count 14
net.stefanhauser.Printy.GUI.Config.BreakMode enumeration 46	net.stefanhauser.Printy.GUI.Config.Errors.CreateException 15
net.stefanhauser.Printy.GUI.Config.BreakRuleControl 6	net.stefanhauser.Printy.GUI.Config.Errors.errMsgFormat 14
net.stefanhauser.Printy.GUI.Config.BreakRuleControl.BreakR uleControl 7	net.stefanhauser.Printy.GUI.Config.Errors.Error 15
net.stefanhauser.Printy.GUI.Config.BreakRuleControl.Dispose 7	net.stefanhauser.Printy.GUI.Config.Errors.ErrorCount 16
net.stefanhauser.Printy.GUI.Config.BreakRuleControl.GetCre atedRule 8	net.stefanhauser.Printy.GUI.Config.Errors.Errors 15
net.stefanhauser.Printy.GUI.Config.Buffer 8	net.stefanhauser.Printy.GUI.Config.Errors.Exception 15
net.stefanhauser.Printy.GUI.Config.Buffer.Buffer 9	net.stefanhauser.Printy.GUI.Config.Errors.SemErr 15
net.stefanhauser.Printy.GUI.Config.Buffer.Close 9	net.stefanhauser.Printy.GUI.Config.Errors.SynErr 16
net.stefanhauser.Printy.GUI.Config.Buffer.EOF 9	net.stefanhauser.Printy.GUI.Config.ErrorType 46
net.stefanhauser.Printy.GUI.Config.Buffer.GetString 9	net.stefanhauser.Printy.GUI.Config.ErrorType enumeration 46
net.stefanhauser.Printy.GUI.Config.Buffer.Peek 9	net.stefanhauser.Printy.GUI.Config.IRuleControl 45
net.stefanhauser.Printy.GUI.Config.Buffer.Pos 9	net.stefanhauser.Printy.GUI.Config.IRuleControl.GetCreatedR ule 45
net.stefanhauser.Printy.GUI.Config.Buffer.Read 9	net.stefanhauser.Printy.GUI.Config.IRuleControl.Text 46
net.stefanhauser.Printy.GUI.Config.Buffer.ClassControl 9	net.stefanhauser.Printy.GUI.Config.Parser 16
net.stefanhauser.Printy.GUI.Config.ClassControl.ClassControl 11	net.stefanhauser.Printy.GUI.Config.Parser.errors 20
net.stefanhauser.Printy.GUI.Config.ClassControl.Dispose 11	net.stefanhauser.Printy.GUI.Config.Parser.la 20
net.stefanhauser.Printy.GUI.Config.ClassControl.GetCreatedR ule 11	net.stefanhauser.Printy.GUI.Config.Parser.Parse 21
net.stefanhauser.Printy.GUI.Config.ConfigForm 11	net.stefanhauser.Printy.GUI.Config.Parser.Parser 20
net.stefanhauser.Printy.GUI.Config.ConfigForm.AdjustedPara meter 13	net.stefanhauser.Printy.GUI.Config.Parser.Parser.Parser 20
net.stefanhauser.Printy.GUI.Config.ConfigForm.ConfigFileNa me 13	net.stefanhauser.Printy.GUI.Config.Parser.Parser.Rule 20
	net.stefanhauser.Printy.GUI.Config.Parser.Parser.scanner 20
	net.stefanhauser.Printy.GUI.Config.Parser.SemErr 21
	net.stefanhauser.Printy.GUI.Config.Parser.t 20
	net.stefanhauser.Printy.GUI.Config.ParserError 21
	net.stefanhauser.Printy.GUI.Config.ParserError.Column 22
	net.stefanhauser.Printy.GUI.Config.ParserError.Line 22
	net.stefanhauser.Printy.GUI.Config.ParserError.ParserError 22
	net.stefanhauser.Printy.GUI.Config.ParserError.Text 22

net.stefanhauser.Printy.GUI.Config.ParserError.ToString	22	net.stefanhauser.Printy.GUI.Config.RuleManager	30
net.stefanhauser.Printy.GUI.Config.ParserError.Type	22	net.stefanhauser.Printy.GUI.Config.RuleManager.Add	32
net.stefanhauser.Printy.GUI.Config.ParserException	23	net.stefanhauser.Printy.GUI.Config.RuleManager.FileName	32
net.stefanhauser.Printy.GUI.Config.ParserException.ErrorCount	23	net.stefanhauser.Printy.GUI.Config.RuleManager.FileNameChanged	33
net.stefanhauser.Printy.GUI.Config.ParserException.ErrorList	24	net.stefanhauser.Printy.GUI.Config.RuleManager.Remove	32
net.stefanhauser.Printy.GUI.Config.ParserException.ParserException	23	net.stefanhauser.Printy.GUI.Config.RuleManager.RemoveAll	32
net.stefanhauser.Printy.GUI.Config.Rule	24	net.stefanhauser.Printy.GUI.Config.RuleManager.RuleAdded	33
net.stefanhauser.Printy.GUI.Config.Rule.Add	26	net.stefanhauser.Printy.GUI.Config.RuleManager.RuleManager	31
net.stefanhauser.Printy.GUI.Config.Rule.BreakMode	25	net.stefanhauser.Printy.GUI.Config.RuleManager.RuleRemoved	33
net.stefanhauser.Printy.GUI.Config.Rule.CreateBitSet	26	net.stefanhauser.Printy.GUI.Config.RuleManager.TempFileName	33
net.stefanhauser.Printy.GUI.Config.Rule.Equals	26	net.stefanhauser.Printy.GUI.Config.RuleManager.WriteToFile	32
net.stefanhauser.Printy.GUI.Config.Rule.FirstSymbol	25	net.stefanhauser.Printy.GUI.Config.RuleType	47
net.stefanhauser.Printy.GUI.Config.Rule.FurtherText	25	net.stefanhauser.Printy.GUI.Config.RuleType enumeration	47
net.stefanhauser.Printy.GUI.Config.Rule.IsValid	27	net.stefanhauser.Printy.GUI.Config.Scanner	33
net.stefanhauser.Printy.GUI.Config.Rule.Number	25	net.stefanhauser.Printy.GUI.Config.Scanner.buffer	37
net.stefanhauser.Printy.GUI.Config.Rule.SecondSymbol	25	net.stefanhauser.Printy.GUI.Config.Scanner.Peek	37
net.stefanhauser.Printy.GUI.Config.Rule.Text	25	net.stefanhauser.Printy.GUI.Config.Scanner.ResetPeek	37
net.stefanhauser.Printy.GUI.Config.Rule.ToString	27	net.stefanhauser.Printy.GUI.Config.Scanner.Scan	38
net.stefanhauser.Printy.GUI.Config.Rule.Type	26	net.stefanhauser.Printy.GUI.Config.Scanner.Scanner	37
net.stefanhauser.Printy.GUI.Config.RuleControl	27	net.stefanhauser.Printy.GUI.Config.SpaceRuleControl	38
net.stefanhauser.Printy.GUI.Config.RuleControl.ActiveControl	28	net.stefanhauser.Printy.GUI.Config.SpaceRuleControl.Dispose	39
net.stefanhauser.Printy.GUI.Config.RuleControl.Dispose	28	net.stefanhauser.Printy.GUI.Config.SpaceRuleControl.GetCreatedRule	39
net.stefanhauser.Printy.GUI.Config.RuleControl.Enabled	29	net.stefanhauser.Printy.GUI.Config.SpaceRuleControl.SpaceRuleControl	39
net.stefanhauser.Printy.GUI.Config.RuleControl.RuleControl	28	net.stefanhauser.Printy.GUI.Config.Symbol	47
net.stefanhauser.Printy.GUI.Config.RuleControl.SetupHelpProvider	28	net.stefanhauser.Printy.GUI.Config.Symbol enumeration	47
net.stefanhauser.Printy.GUI.Config.RuleControlFactory	29	net.stefanhauser.Printy.GUI.Config.Token	39
net.stefanhauser.Printy.GUI.Config.RuleControlFactory.CreateRuleControl	29	net.stefanhauser.Printy.GUI.Config.Token.col	40
net.stefanhauser.Printy.GUI.Config.RuleEventArgs	29	net.stefanhauser.Printy.GUI.Config.Token.kind	40
net.stefanhauser.Printy.GUI.Config.RuleEventArgs.Rule	30	net.stefanhauser.Printy.GUI.Config.Token.line	40
net.stefanhauser.Printy.GUI.Config.RuleEventArgs.RuleEventArgs	30	net.stefanhauser.Printy.GUI.Config.Token.next	40
net.stefanhauser.Printy.GUI.Config.RuleEventHandler	48		
net.stefanhauser.Printy.GUI.Config.RuleEventHandler type	48		

net.stefanhauser.Printy.GUI.Config.Token.pos 41
 net.stefanhauser.Printy.GUI.Config.Token.val 41
 net.stefanhauser.Printy.GUI.Config.Utilities 41
 net.stefanhauser.Printy.GUI.Config.Utilities.BreakRules 41
 net.stefanhauser.Printy.GUI.Config.Utilities.Rules 42
 net.stefanhauser.Printy.GUI.Config.Utilities.SingleSymbol 42
 net.stefanhauser.Printy.GUI.Config.Utilities.SymbolAll 42
 net.stefanhauser.Printy.GUI.Config.Utilities.SymbolAllIndent 42
 net.stefanhauser.Printy.GUI.Config.Wrapper 42
 net.stefanhauser.Printy.GUI.Config.Wrapper.BreakMode 43, 44
 net.stefanhauser.Printy.GUI.Config.Wrapper.RuleType 43, 45
 net.stefanhauser.Printy.GUI.Config.Wrapper.Symbol 43, 45
 net.stefanhauser.Printy.GUI.Config.Wrapper.ToString 44
 net.stefanhauser.Printy.GUI.Config.Wrapper.Wrapper 44

P

Parser class 16
 errors 20
 la 20
 Parse 21
 Parser 20
 Rules 20
 scanner 20
 SemErr 21
 t 20
 Parser.cs 51
 ParserError class 21
 Column 22
 Line 22
 ParserError 22
 Text 22
 ToString 22
 Type 22
 ParserException class 23
 ErrorCount 23
 ErrorList 24
 ParserException 23

R

Rule class 24
 Add 26
 BreakMode 25
 CreateBitSet 26
 Equals 26
 FirstSymbol 25
 FurtherText 25
 IsValid 27
 Number 25
 SecondSymbol 25
 Text 25
 ToString 27
 Type 26
 Rule.cs 52
 RuleControl class 27
 ActiveControl 28
 Dispose 28
 Enabled 29
 RuleControl 28
 SetupHelpProvider 28
 RuleControl.cs 52
 RuleControlFactory class 29
 CreateRuleControl 29
 RuleControlFactory.cs 52
 RuleEventArgs class 29
 Rule 30
 RuleEventArgs 30
 RuleManager class 30
 Add 32
 Filename 32
 FileNameChanged 33
 Remove 32
 RemoveAll 32
 RuleAdded 33
 RuleManager 31
 RuleRemoved 33
 TempFileName 33
 WriteToFile 32

RuleManager.cs 53

Wrapper 44

Wrapper.cs 54

S

Scanner class 33

buffer 37

Peek 37

ResetPeek 37

Scan 38

Scanner 37

Scanner.cs 53

SpaceRuleControl class 38

Dispose 39

GetCreatedRule 39

SpaceRuleControl 39

SpaceRuleControl.cs 54

T

Token class 39

col 40

kind 40

line 40

next 40

pos 41

val 41

U

Utilities class 41

BreakRules 41

Rules 42

SingleSymbol 42

SymbolAll 42

SymbolAllIndent 42

Utilities.cs 54

W

Wrapper class 42

BreakMode 43, 44

RuleType 43, 45

Symbol 43, 45

ToString 44