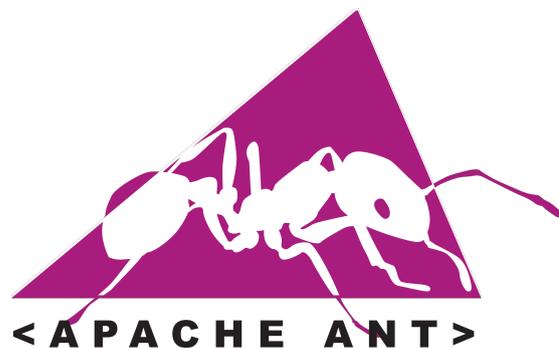


# ANT-Worker Tasks

Alpha-Version



MICHAEL NIEDERMAIR

2011-05-01 (Revision: 268)

# Inhaltsverzeichnis

1	Einführung .....	4
1.1	Version .....	4
1.2	Aufbau .....	4
1.3	Die Task-Gruppen .....	5
1.4	Einbinden der Bibliotheken .....	5
1.4.1	Beispiel: Worker-ipdf .....	5
2	Klassen-Struktur .....	7
2.1	Die Basic-Tasks .....	8
2.2	Die Gruppen-Tasks .....	8
2.3	Arbeitsverzeichnis .....	9
2.4	Quell - Zielverzeichnis .....	10
2.5	PDF-Tasks .....	11
2.6	T <sub>E</sub> X-Tasks .....	12
3	Ausführen externer Programme .....	13
3.1	Welches Betriebssystem verwende ich .....	13
3.2	Aufrufe definieren .....	13
4	ANT-Worker-Task .....	16
4.1	Basic-Tasks .....	16
4.1.1	foreach .....	16
4.1.2	for .....	18
4.1.3	ifchecksum .....	19
4.1.4	ifdiff .....	21
4.1.5	ifnewer .....	22
4.2	Gruppendefinition .....	23
4.2.1	defgroup .....	23
4.2.2	group .....	24
4.3	Datei-Tools .....	25
4.3.1	copynamespace .....	25
4.3.2	copyreplace .....	26
4.3.3	copysubdir .....	27
4.3.4	removeother .....	28
4.3.5	remove .....	29
4.4	Konfiguration .....	30
4.4.1	configexecute .....	30
4.4.2	configkv .....	31
4.4.3	variable .....	32
4.4.4	executecmd .....	33
4.5	PDF-Bearbeitung .....	34
4.5.1	pdfburst .....	34
4.5.2	pdfcompress .....	35
4.5.3	pdfcroppage .....	36
4.5.4	pdfinsert .....	38
4.5.5	pdfmerge .....	40
4.5.6	pdfmetadata .....	42
4.5.7	pdfnup .....	43
4.5.8	pdfremovinglaunchaction .....	44
4.5.9	pdfrotate .....	45
4.5.10	pdfsecure .....	46
4.5.11	pdfwatermark .....	48

4.5.12	pages .....	49
4.5.13	pdffilepages .....	50
4.6	Utility .....	51
4.6.1	systeminfo .....	51
4.6.2	timer .....	52
5	Nutzungsrechte, Lizenz und Sourcecode .....	53
5.1	Apache-ANT .....	53
5.2	Worker-Base .....	54
5.3	Worker-ipdf .....	54
5.4	Worker-Tools .....	55
6	ToDo .....	56
	Übersicht Ant-Task .....	57
	Übersicht externe Programme .....	58
	Übersicht JAR-Archive .....	59
	Übersicht Klassen / Interfaces .....	60
	Stichwortverzeichnis .....	61

# 1 Einführung

Die Worker-Tasks sind eine Sammlung von Apache-ANT-Erweiterungen, die einem das Leben mit dem Erstellen von großen Dokumenten erleichtern soll. Mit der Zeit sind viele Funktionen dazu gekommen, die nicht direkt mit der Dokumentenerstellung zu tun haben.

Apache-ANT<sup>1</sup> ist ein build-System, ähnlich wie das gute alte *make* aus Unix und ist in Java programmiert. Ursprünglich diente es dazu, java-Dateien entsprechend bestimmter Abhängigkeiten zu übersetzen. Inzwischen gibt es eine Vielzahl von weiteren Funktionen, die über das reine Quellcodeübersetzen weit hinaus gehen.

Mehr zu Apache-ANT findet man unter <http://ant.apache.org/>.

Aus lizenzrechtlichen Gründen sind die einzelnen Funktionen in Pakete aufgeteilt, je nachdem, welche Bibliothek diese nutzen. Viele der verwendeten Bibliotheken verwenden eine Lizenz, die vorschreibt, welche Lizenz das entsprechende Paket haben muss, daher für manchen etwas komplizierter.

## 1.1 Version

Die Worker-Tasks setzen folgende Software voraus:

- Java 6 (jdk 1.6)  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Apache-ANT 1.8.x  
<http://ant.apache.org/>

## 1.2 Aufbau

Nachfolgende Abbildung 1.1 zeigt schematisch, wie Apache-ANT die entsprechenden Erweiterungen nutzt.

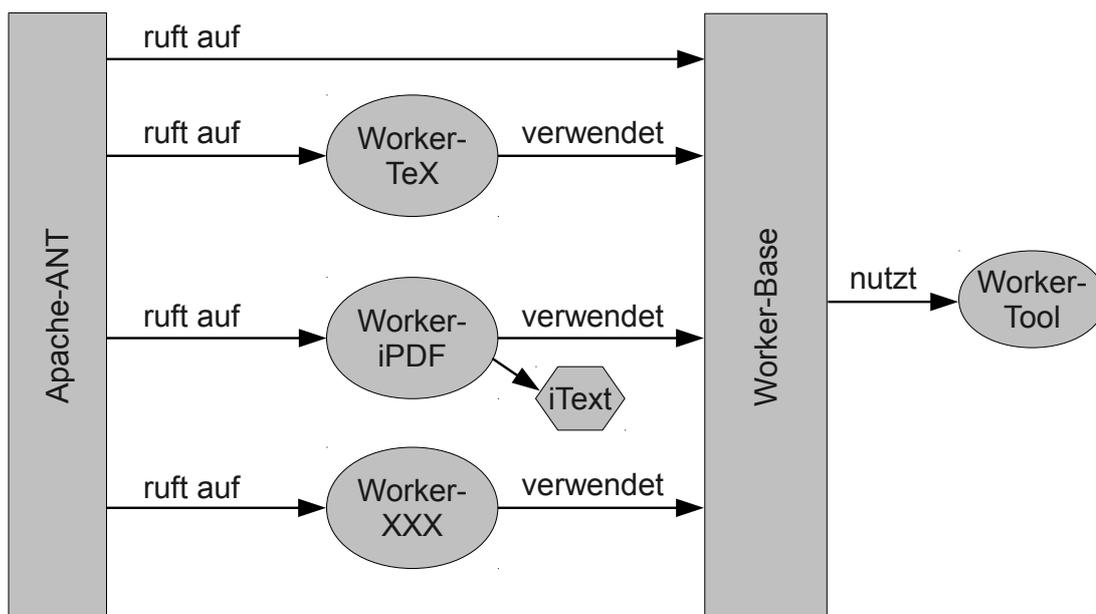


Abb. 1.1: Aufbau

<sup>1</sup> Apache License 2.0

Dabei bindet Apache-ANT die jeweilige Erweiterung ein und ruft den entsprechenden Task auf. Der Task verwendet bei Bedarf eine entsprechende externe Bibliothek oder ruft dabei ein externes Programm auf. Unterstützt wird der Task dabei von der Worker-Base und den Worker-Tools.

## 1.3 Die Task-Gruppen

Worker-Base	Basisfunktionalität und Unterstützerklassen für andere Tasks <i>ifnewer, ifdiff, parallelqueue, removeother, remove, timer, defgroup, group, ...</i>
Worker-TeX	Tasks für $\TeX$ , $\LaTeX$ , ( <i>latex, xelatex, lualatex, makeindex, ...</i> )
Worker-ipdf	Tasks für die PDF-Bearbeitung <i>pdfburst, pdfsecure, pdfinsert, pdfmerge, pdfnup, pdfcroppage, ...</i>
...	weitere sind in Entwicklung

Eine komplette Beschreibung für alle Tasks findet sich ab Abschnitt [4 auf Seite 16](#) und eine Übersicht auf Seite [57](#).

## 1.4 Einbinden der Bibliotheken

Apache-ANT bietet drei Möglichkeiten, Bibliotheken einzubinden:

1. Alle Bibliotheken (jar-Archive mit der Dateiendung *jar*) werden in das lib-Verzeichnis von Apache-ANT kopiert. Dies funktioniert nur, wenn man auch die entsprechenden Rechte besitzt!
2. Man verwendet die Umgebungsvariable CLASSPATH und verweist damit auf die entsprechenden jar-Archive. Mit `taskdef` werden dann die Tasks definiert.
3. Man bindet die Tasks und entsprechenden Bibliotheken über `taskdef` und dem Attribut `classpath` bzw. `classpathref` ein. Dazu ist es sinnvoll, wenn man alle jar-Archive über `path` und ein entsprechendes `fileset` lädt.

### 1.4.1 Beispiel: Worker-ipdf

In Abschnitt [4.5 auf Seite 34](#) werden die Tasks für die PDF-Bearbeitung beschrieben.

Für das Grundsystem werden die beiden Bibliotheken `worker-base.jar` und `worker-tools.jar` benötigt. Der Task `pdfsecure` ist in der Bibliothek `worker-ipdf.jar` enthalten und bindet als externe Jars `itext.jar`, `bcmail-jdk16-145.jar`, `bcprov-jdk16-145.jar` und `bctsp-jdk16-145.jar` ein.

Nun sollen alle PDF-Dateien aus einem bestimmten Verzeichnis verschlüsselt und in einem Zielverzeichnis abgelegt werden. Ist die verschlüsselte PDF-Datei im Zielverzeichnis neuer als im Quellverzeichnis, soll nichts passieren. Damit alles etwas schneller geht, sollen pro CPU-Kern zwei Threads parallel ausgeführt werden.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- $Id: build.xml 249 2011-03-18 20:31:47Z michael $ -->
3 <project name="Worker-Test mit iPDF" basedir="." default="pdfs">
4
5     <!-- hier liegen alles jar-Archive, die eingebunden werden sollen -->
6     <path id="jars">
7         <fileset dir="./lib">
8             <include name="*.jar" />
9         </fileset>
10    </path>
11
12    <!-- Die Tasks von Worker-base werden eingebunden -->
13    <taskdef resource="de/nm/ant/antlib-base.xml" classpathref="jars" />
14    <!-- Die Tasks von Worker-ipdf werden eingebunden -->
15    <taskdef resource="de/nm/ant/antlib-ipdf.xml" classpathref="jars" />
16
17    <!-- Variable für das Ziel-Verzeichnis -->
18    <property name="target.dir" value="target" />
19
20    <!-- Initialisierung -->
21    <target name="init" description="Init">

```

## 6 1 Einführung

```
22         <!-- Zielverzeichnis erstellen, wenn es noch nicht da ist -->
23         <mkdir dir="${target.dir}" />
24     </target>
25
26     <!-- alles wieder löschen -->
27     <target name="clean">
28         <delete dir="${target.dir}" />
29     </target>
30
31     <!-- PDFs verschlüsseln -->
32     <target name="pdfs" depends="init">
33         <ifnewer threadsperprocessor="2"
34             srcdir="pdfs"
35             destdir="${target.dir}"
36             srcextname="*\ pdf$"
37             destextname="pdf">
38             <pdfsecure />
39         </ifnewer>
40     </target>
41
42 </project>
```

**Listing 1.1:** [build.xml](#)

Dabei wird zuerst in Zeile 6 ein path definiert, der die entsprechenden Archive enthält. Damit wir nichts vergessen und nicht zu viel tippen müssen, binden wir alle jar-Archive aus dem Verzeichnis ein.

Für die beiden Pakete (*worker-base* und *worker-ipdf*) werden die Definitionen eingebunden, mit einem Verweis, wo diese zu finden sind (in den jar-Archiven).

Für das Zielverzeichnis wird in Zeile 18 eine Variable definiert.

Anschließend wird der Task `init` definiert, der sicher stellt, dass das Verzeichnis `target` auch existiert. Der Task `clean` sorgt dafür, dass alle erstellten PDF-Dateien wieder gelöscht werden.

In Zeile 32 wird der eigentliche Task definiert, der die PDFs verschlüsselt. Mit dem Task `ifnewer` wird das Quellverzeichnis mit dem Zielverzeichnis verglichen. Ist das Ziel nicht vorhanden oder älter, so wird der eingebundene Task `pdfsecure` aufgerufen. Damit man hier nicht zu viel tippen muss, übergibt der Task `ifnewer` die entsprechenden Parameter an den eingebundenen Task. Über das Attribut `threadsperprocessor` wird festgelegt, dass pro CPU-Kern zwei Threads parallel laufen. Bei meinem Testsystem mit zwei Kernen werden somit vier PDF-Dateien parallel verschlüsselt.

Ein Aufruf könnte folgende Ausgabe ergeben (je nach Anzahl der CPU-Kerne):

```
> ant
Buildfile: /raid/daten/sv/worker/tests/build.xml

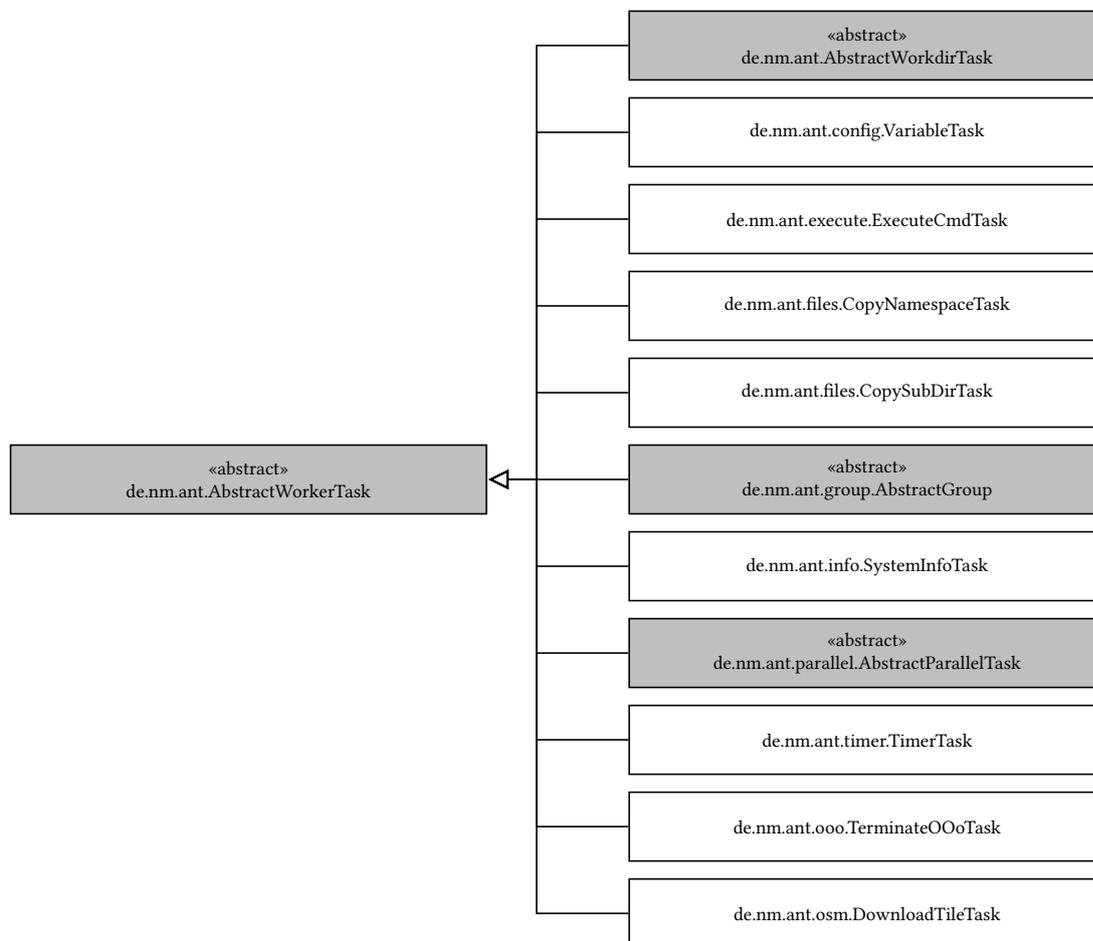
init:
    [mkdir] Created dir: /raid/daten/sv/worker/tests/target

pdfs:
    [ifnewer] a4_2.pdf
    [pdfsecure] a4_2.pdf
    [ifnewer] a4.pdf
    [ifnewer] a4_3.pdf
    [pdfsecure] a4.pdf
    [ifnewer] ie.pdf
    [pdfsecure] a4_3.pdf
    [ifnewer] i2.pdf
    [ifnewer] a4_4.pdf
    [ifnewer] a4_1.pdf
    [ifnewer] s4.pdf
    [pdfsecure] ie.pdf
    [pdfsecure] i2.pdf
    [pdfsecure] a4_4.pdf
    [pdfsecure] a4_1.pdf
    [pdfsecure] s4.pdf

BUILD SUCCESSFUL
Total time: 2 seconds
```

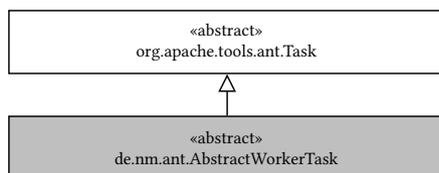
## 2 Klassen-Struktur

Die abstrakte Basisklasse für alle ANT-Worker-Tasks ist `AbstractWorkerTask`.



**Abb. 2.1:** AbstractWorkerTask

Diese ist von der Apache-ANT Task-Klasse abgeleitet und stellt somit alle notwendigen Methoden bereit, so dass der Task in der build-Datei verwendet werden kann.



**Abb. 2.2:** von Task abgeleitet

## 2.1 Die Basic-Tasks

Die wesentlichen Tasks sind:

- `ifnewer` (Klasse `IfNewerTask`)  
Führt Aktionen aus, wenn die Zielfeile älter oder nicht vorhanden ist.
- `ifdiff` (Klasse `IfDiffTask`)  
Führt Aktionen aus, wenn sich die Quelldatei gegenüber der Zielfeile verändert hat (es wird eine Differenz gebildet).
- `ifchecksum` (Klasse `IfChecksumTask`)  
Führt Aktionen aus, wenn sich die Prüfsumme (crc32, adler32, md5 oder sha-256) der Datei geändert hat.

Alle drei Klassen sind von `AbstractParallelTask` abgeleitet.

Zusätzlich stellt die Klasse `ParallelQueueTask` den Task `parallelqueue` zur Verfügung, welche es ermöglicht, dass alle eingebetteten Paralleltasks die selbe Queue zur parallelen Ausführung nutzen.

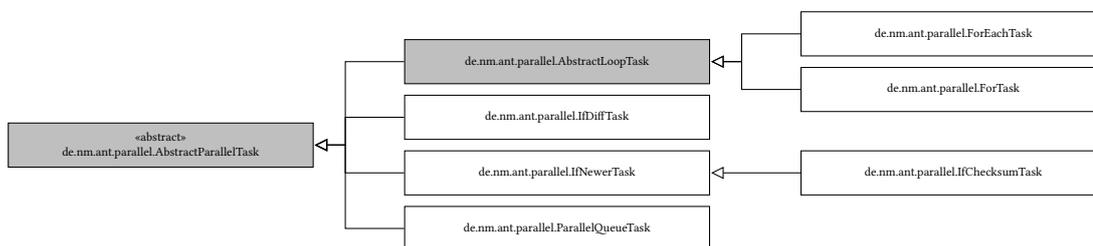


Abb. 2.3: `AbstractParallelTask`

## 2.2 Die Gruppen-Tasks

Damit man zusammenhängende Tasks immer wieder verwenden kann (mit anderen Aufrufparametern), stellt die Klasse `DefGroupTask` den Task `defgroup` zur Verfügung. Dieser sammelt eingebettete Tasks und speichert diese. Der Task `group` (Klasse `GroupTask`) führt diese aus, wobei bei Bedarf entsprechende Parameter weitergegeben werden.

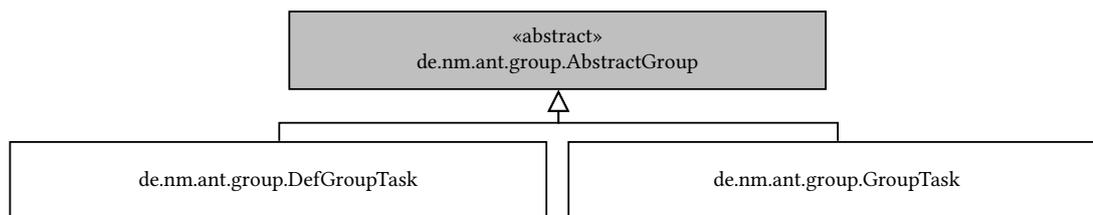
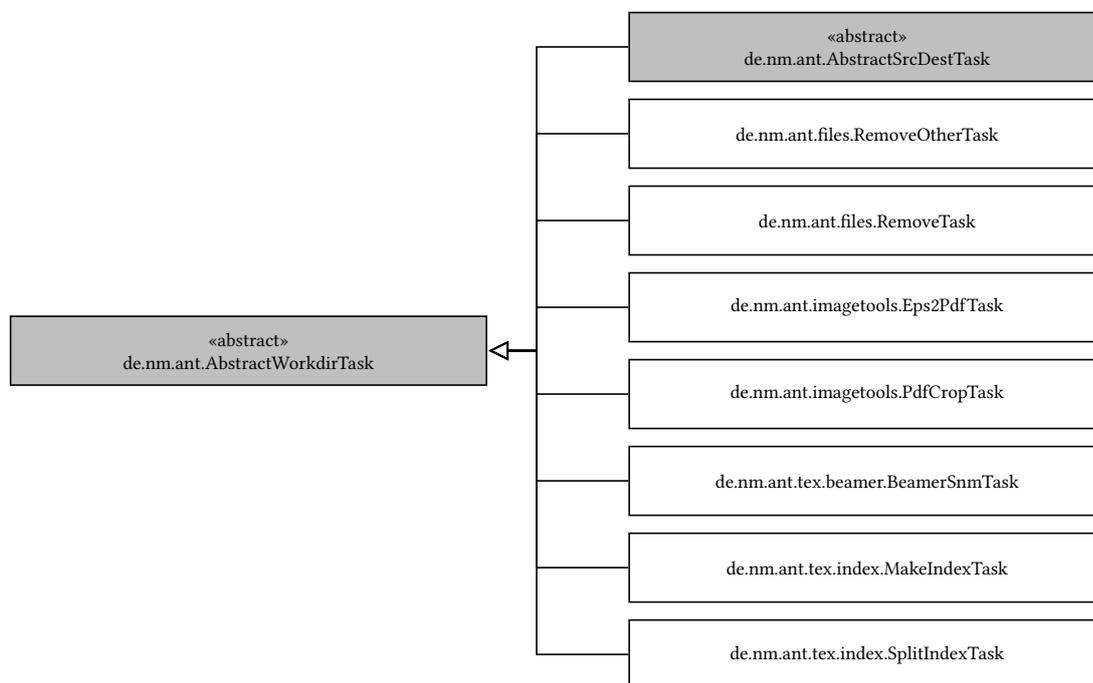


Abb. 2.4: `AbstractGroup`

## 2.3 Arbeitsverzeichnis

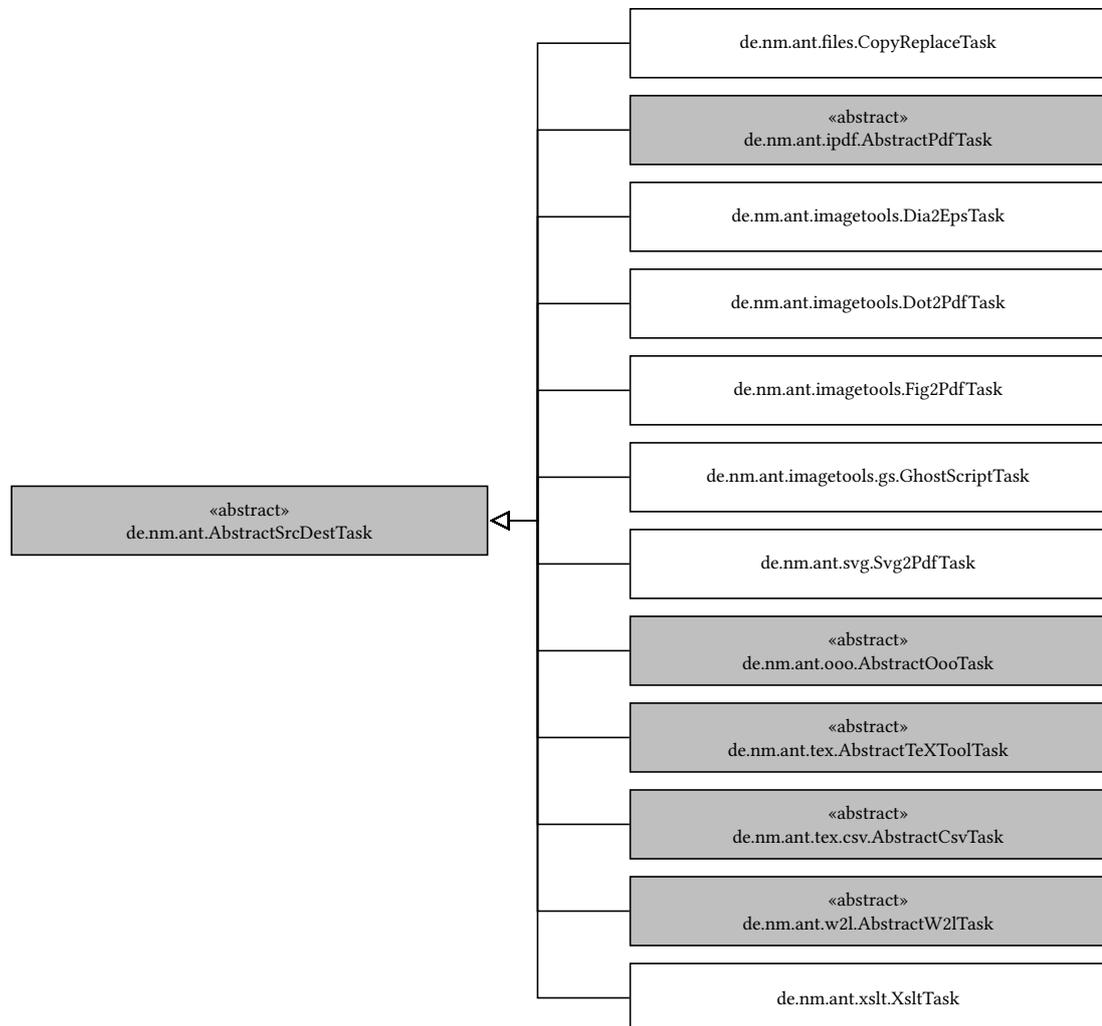
Die Klasse `AbstractWorkdirTask` stellt die Funktionalität für das Arbeiten mit einem Arbeitsverzeichnis zur Verfügung.



**Abb. 2.5:** `AbstractWorkdirTask`

## 2.4 Quell - Zielverzeichnis

Für Tasks, die eine Datei im Quellverzeichnis mit einer Datei im Zielverzeichnis vergleichen bzw. übersetzen, steht die Klasse `AbstractSrcDestTask` zur Verfügung.



**Abb. 2.6:** `AbstractSrcDestTask`

## 2.5 PDF-Tasks

Speziell für die PDF-Bearbeitung mit der Bibliothek iText stehen einige Tasks zur Verfügung, die von der Klasse `AbstractPdfTask` erben. Soll ein Task nur bestimmte Seiten bearbeiten, so wird zusätzlich die Klasse `AbstractPagesPdfTask` verwendet.

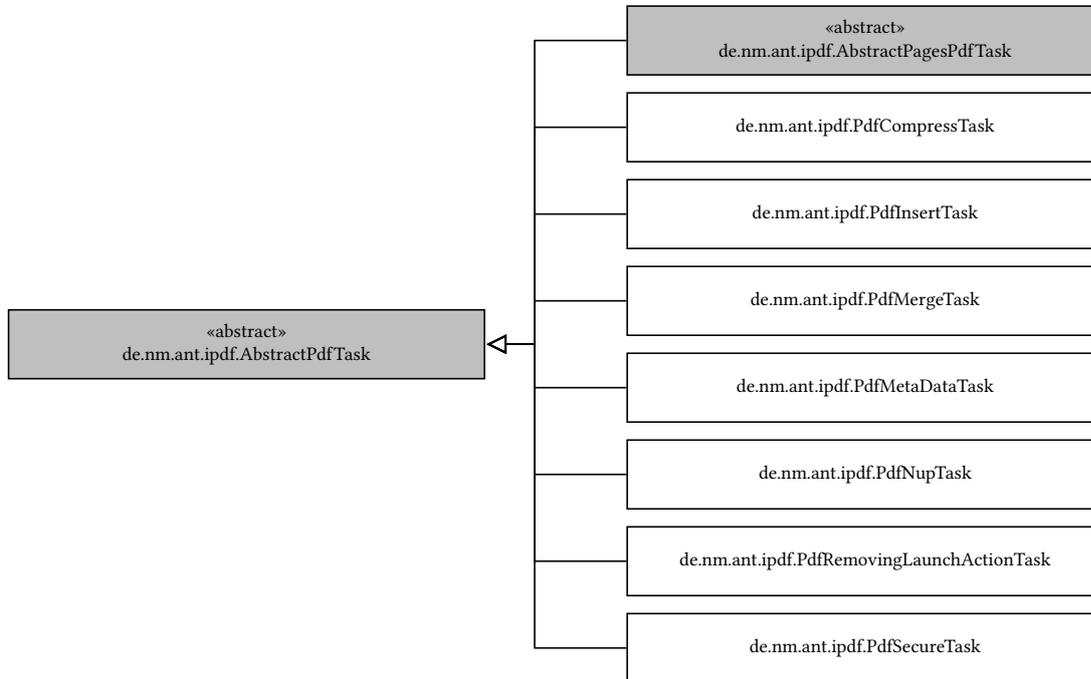


Abb. 2.7: AbstractPdfTask

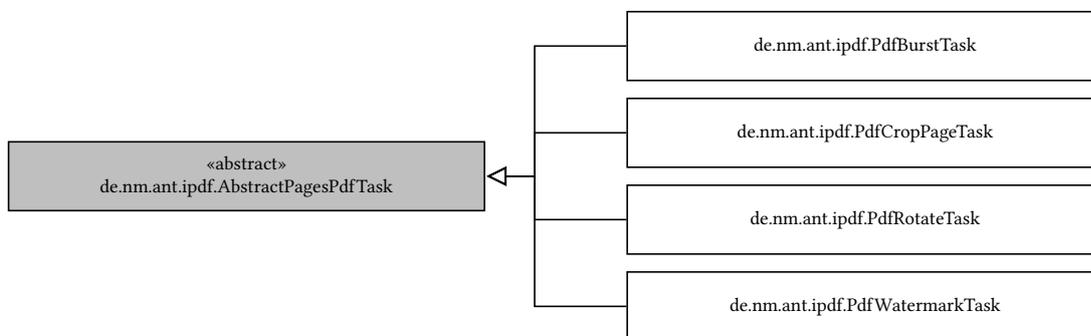
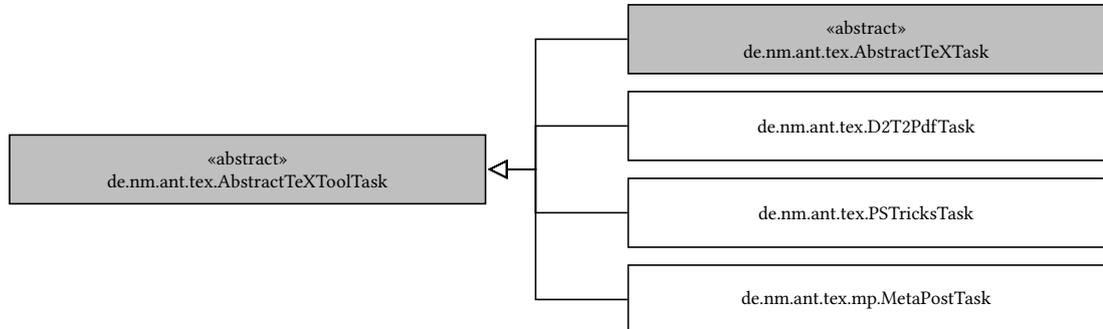


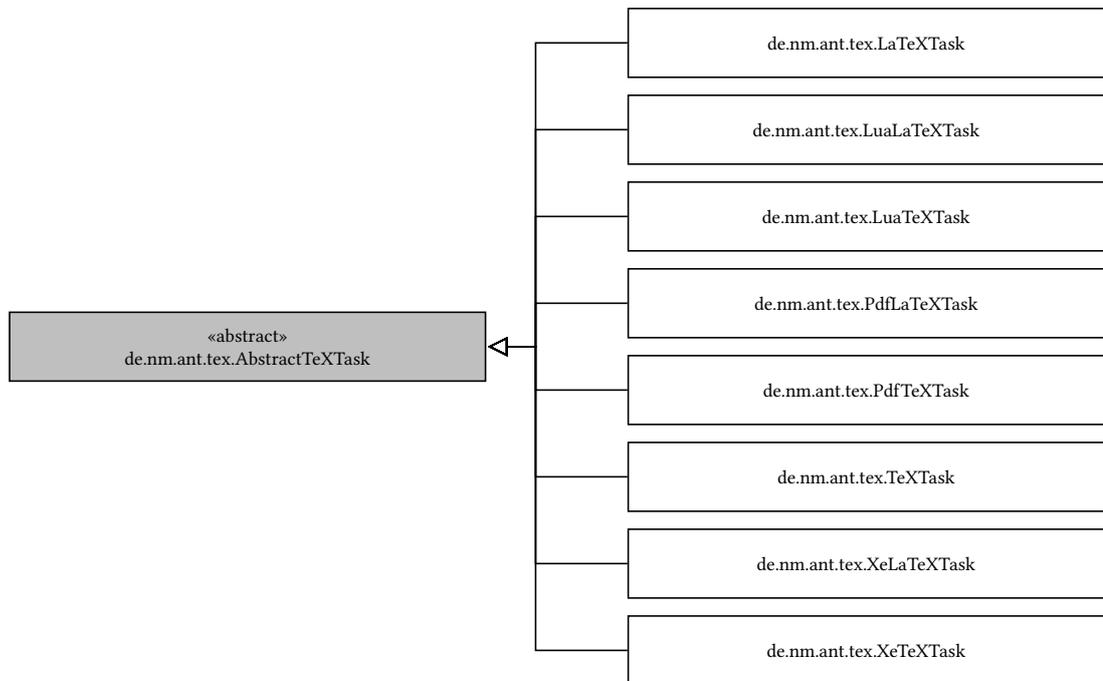
Abb. 2.8: AbstractPagesPdfTask

## 2.6 T<sub>E</sub>X-Tasks

Die T<sub>E</sub>X-Tasks stellen ursprünglich die Entwicklungsgrundlage der ANT-Worker-Task dar. Inzwischen sind diese aber nur einige Tasks unter vielen. Die Basisklasse ist dabei für alle T<sub>E</sub>X-Tools die Klasse `AbstractTeXToolTask` und für die eigentlichen T<sub>E</sub>X- L<sup>A</sup>T<sub>E</sub>X-Programme die Klassen `AbstractTeXTask`.



**Abb. 2.9:** `AbstractTeXToolTask`



**Abb. 2.10:** `AbstractTeXTask`

## 3 Ausführen externer Programme

Viele Tasks führen externe Programme aus. Das Ausführen externer Programme ist je nach Betriebssystem nicht immer einfach, da diverse Programme auf Umgebungsvariablen angewiesen sind. Jedoch kann es sein, dass diese an den externen Aufruf des Betriebssystems nicht weitergereicht werden. Einige Systeme unterscheiden auch zwischen System-Umgebungsvariablen und Benutzer-Umgebungsvariablen, die unterschiedlich verwendet werden oder starten bei einem externen Aufruf eine komplett neue Umgebung.

Die Tasks werden primär unter Linux entwickelt. Linux stellt im Normalfall für alle Programme die Startdatei im Verzeichnis `/usr/bin` bzw. `/usr/local/bin` zur Verfügung. Alternativ kann hier auch ein symbolischer Link verwendet werden. Dies hat den Nebeneffekt, dass nicht für jedes Programm der Suchpfad (Umgebungsvariable `PATH`) ergänzt werden muss. Somit kann das Programm nur mit seinem Namen ohne eine Pfadangabe aufgerufen werden. Die Tasks rufen daher das Programm auch nur mit seinem Namen auf. Wird unter Linux, z. B. mit der `bash` eine Umgebungsvariable mit `export` definiert, so wird diese an alle Kindprozesse weitergereicht.

Ist das entsprechende Programm nicht im Suchpfad enthalten und möchte man das Verzeichnis nicht ständig hinzufügen oder gar das Programm in einem Skript ausführen, so kann für jeden externen Aufruf eine eigene Umgebung definiert werden.

### 3.1 Welches Betriebssystem verwende ich

Java ermittelt den Namen des Betriebssystems und stellt diesen unter den Systemproperties mit dem Namen `os.name` zur Verfügung. Mit dem Task `systeminfo` können alle Systemproperties ausgegeben werden. Verwendet man das Attribut `filter`, so werden nur die Werte ausgegeben, die dem Filter entsprechen.

```
<systeminfo verbose="true" filter="os.name" />
```

Als Ausgabe erhält man z. B.

Linux:

```
[systeminfo] os.name=Linux
```

Windows 7:

```
[systeminfo] os.name=Windows 7
```

### 3.2 Aufrufe definieren

Mit dem Task `executecmd` werden die Befehle und Umgebungen für einen externen Aufruf festgelegt. Vor dem Ausführen eines externen Programms (egal welcher Task verwendet wird), wird in der Konfiguration von `executecmd` nachgesehen, ob für das Programm ein Eintrag vorhanden ist. Ist dies der Fall, so werden die eingetragenen Einstellungen verwendet.

```
<executecmd>
  <configexecute name="xelatex" osname="Linux.*" params="xelatex" shellcmd="bash">
    <env key="TEXMFHOME" value="/home/mgn/texmf" />
    # Dies ist ein Skript
    # ... weitere Befehle
  </configexecute>
  <configexecute name="xelatex" osname="Windows.*" params="xelatex" shellcmd="cmd /C
">
    <env key="TEXMFHOME" value="/home/mgn/texmf" />
    rem Dies ist ein Skript
    rem ... weitere Befehle
  </configexecute>
  <configexecute name="dot2tex" osname="Windows.*"
    params="C:\Programme\Python27\python.exe
      c:\Programme\Python27\Lib\site-packages\dot2tex\dot2tex.py" />
  <configexecute name="dia" osname="Windows.*"
    params="'C:\Program Files (x86)\Dia\bin\dia.exe'" />
```

## 14 3 Ausführen externer Programme

```
<configexecute name="dot" osname="Windows.*"
  params="'C:\Program Files (x86)\Graphviz2.27\bin\dot.exe' " />
</executecmd>
```

Der Task `configexecute` definiert für das Programm `xelatex` unter dem Betriebssystem Linux (es wird hier über einen regulären Ausdruck verglichen), welches in einem Skript ausgeführt wird (mit der Shell `bash`). Unter Windows wird das Skript über `cmd` ausgeführt. Tritt kein Fehler bei der Ausführung auf, so wird das Skript gleich wieder gelöscht. Soll es erhalten bleiben, so kann bei dem ausführenden Task das Attribut `deletescriptfile` verwendet werden.

Wird kein Text im Task angegeben, so wird das Programm ohne einen Shell-Aufruf gestartet. Eventuelle Umgebungsvariablen werden dabei bei dem Aufruf mit angegeben.

Bei den angegebenen Parametern wird jeder Parameter durch ein Leerzeichen getrennt, wie es bei der Shell üblich ist. Ist z. B. ein Leerzeichen im Dateinamen enthalten, so muss der Parameter in einfache Hochkommata gesetzt werden.

Das Programm `dot2tex` wird unter Linux ganz normal aufgerufen. Dabei wird Python mit dem entsprechenden Pythonskript selbstständig gestartet. Unter Windows funktioniert dies leider so nicht. Hier muss Python direkt aufgerufen werden. Als Parameter muss das entsprechende Skript angegeben werden.

Beispiel: `dot2tex -> pdf`

```
<ifnewer srcdir="src/d2t"
  destdir="$target.dir"
  srcextname="*.d2t$"
  verbose="true"
  destextname="pdf"
  threadsperprocessor="0" >
  <d2t2pdf debug="true" verbose="true" deletescriptfile="false" >
    \documentclass [12pt]{article}
    \usepackage{libertine}
    \usepackage{xcolor}
    \usepackage{pst-all,pst-grad}
    \usepackage[a0paper]{geometry}
    % Styles
    \newsstyle{umlLR}{arrowinset=0,arrowlength=1,arrowscale=3,ArrowFill=false,
      angleA=0,angleB=180,nodesepA=1pt,armA=1cm,armB=1cm}
    \newsstyle{umlBT}{arrowinset=0,arrowlength=1,arrowscale=3,ArrowFill=false,
      angleA=-90,angleB=90,nodesepA=1pt,armA=1cm,armB=1cm}
    \pagestyle{empty}
    \begin{document}%
  </d2t2pdf >
  <pdfcrop />
  <removeother othernames="pdf" />
</ifnewer >
```

Der Task `d2t2pdf` ruft zuerst `dot2tex` auf und erzeugt so eine kleine  $\text{\LaTeX}$ -Datei ohne Präamble etc.

```
\begin{pspicture}[linewidth=1bp](0bp,0bp)(290bp,171bp)
\pstVerb{2 setlinejoin}
\rput(57bp,130bp){\rnode{MindMap}{\psovalbox[] {MindMap}}}
\rput(167bp,152bp){\rnode{Netzplan}{\psovalbox[] {Netzplan}}}
\rput(189bp,43bp){\rnode{Strukturdiagramm}{\psovalbox[] {Strukturdiagramm}}}
\rput(123bp,86bp){\rnode{DOT}{\psovalbox[] {DOT}}}
\rput(80bp,19bp){\rnode{Pfeildiagramm}{\psovalbox[] {Pfeildiagramm}}}
\ncline[]{-}{DOT}{Netzplan}
\ncline[]{-}{DOT}{MindMap}
\ncline[]{-}{DOT}{Pfeildiagramm}
\ncline[]{-}{DOT}{Strukturdiagramm}
\end{pspicture}
```

Der Task erzeugt dann eine neue Datei, mit dem enthaltenen Text und der erzeugten  $\text{\LaTeX}$ -Datei, gefolgt von einem `\end{document}`.

```
\documentclass [12pt]{article}
\usepackage{libertine}
\usepackage{xcolor}
\usepackage{pst-all,pst-grad}
\usepackage[a0paper]{geometry}
% Styles
\newsstyle{umlLR}{arrowinset=0,arrowlength=1,arrowscale=3,ArrowFill=false,
```

```

angleA=0,angleB=180,nodesepA=1pt,armA=1cm,armB=1cm}
\newsstyle{umlBT}{arrowinset=0,arrowlength=1,arrowscale=3,ArrowFill=false,
angleA=-90,angleB=90,nodesepA=1pt,armA=1cm,armB=1cm}
\pagestyle{empty}
\begin{document}%

\begin{pspicture}[linewidth=1bp](0bp,0bp)(290bp,171bp)
\pstVerb{2 setlinejoin}
\rput(57bp,130bp){\rnode{MindMap}{\psovalbox[] {MindMap}}}
\rput(167bp,152bp){\rnode{Netzplan}{\psovalbox[] {Netzplan}}}
\rput(189bp,43bp){\rnode{Strukturdiagramm}{\psovalbox[] {Strukturdiagramm}}}
\rput(123bp,86bp){\rnode{DOT}{\psovalbox[] {DOT}}}
\rput(80bp,19bp){\rnode{Pfeildiagramm}{\psovalbox[] {Pfeildiagramm}}}
\ncline[]{-}{DOT}{Netzplan}
\ncline[]{-}{DOT}{MindMap}
\ncline[]{-}{DOT}{Pfeildiagramm}
\ncline[]{-}{DOT}{Strukturdiagramm}
\end{pspicture}
\end{document}

```

Anschließend wird diese Datei über den externen Aufruf ausgeführt.

```

...
[d2t2pdf] workdir=/home/daten/sv/worker/target
[d2t2pdf] creating /home/daten/sv/worker/target/script_d2t2pdf_000000
[d2t2pdf] bash /home/daten/sv/worker/target/script_d2t2pdf_000000
...

```

Das erzeugte Skript sieht dabei wie folgt aus:

```

TEXMFHOME=/home/mgn/texmf
# Dies ist ein Skript
# ... weitere Befehle

xelatex -interaction=nonstopmode bsp_d2t_01.tex-tmp

```

Anschließend steht dann im Verzeichnis `target` das erzeugte PDF zur Verfügung (mit `pdfcrop` ohne Rand). Alle erzeugten Daten (mit `removeother`) werden bis auf die PDF-Datei gelöscht.

## 4 ANT-Worker-Task

### 4.1 Basic-Tasks

#### 4.1.1 foreach

Der Task stellt eine Schleife über eine Liste oder eine Dateiauswahl dar, der alle eingebetteten Tasks x-mal aufruft. Die verwendete Variable kann dabei bei den eingebetteten Tasks verwendet werden. Aber Achtung: Da die Variable global gehalten wird, kann bei Threads nicht gewährleistet werden, dass diese den korrekten Wert hat, wenn man auf diese über  $\{i\}$  zugreift.

Jar-Archiv                      worker-base.jar

Parameter

<b>delimiter</b>	Der Trenner für die Liste.
<b>full</b>	Legt fest, ob der absolute Pfad der Datei verwendet werden soll oder der relative Pfad bezogen auf das Fileset (Grundeinstellung: nein).
<b>list</b>	Eine Liste von Werten, die durch einen Trenner getrennt ist.
<b>param</b>	Legt den Variablennamen für die Schleife fest (Grundeinstellung i). Die Variable wird nach der Schleife wieder gelöscht.

Parameter geerbt von `de.nm.ant.parallel.AbstractParallelTask`

<b>exitonerror</b>	Legt fest, ob bei einem Fehler der Task abgebrochen werden soll (Grundeinstellung ja).
<b>failonany</b>	Legt fest, ob Tasks, die in der Warteschleife sind, abgebrochen werden sollen (Grundeinstellung nein).
<b>keepalivetime</b>	Legt die Lebensdauer für einen warteten Task fest (Grundeinstellung 60s).
<b>maxpoolsize</b>	Legt fest, wie viele Aufträge in die Warteschlange aufgenommen werden können.
<b>threadcount</b>	Legt die Anzahl an gleichzeitigen Threads fest (Wird von 'threadsperprocessor' überschrieben, wenn dieser Parameter verwendet wird.).
<b>threadsperprocessor</b>	Legt die Anzahl der Threads für einen Prozessor fest. Das System ermittelt die Anzahl an Prozessoren und multipliziert dies mit der Anzahl an Threads pro Prozessors und legt somit die Anzahl der Threads fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

<b>debug</b>	Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).
<b>deletescriptfile</b>	Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).
<b>encoding</b>	Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).
<b>trim</b>	Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).
<b>usenewenvironment</b>	Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).
<b>verbose</b>	Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).
<b>verboseheader</b>	Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

Innertasks

`fileset`                    Legt die Dateien fest, die abgearbeitet werden sollen.

Innertasks geerbt von `de.nm.ant.parallel.AbstractParallelTask`

`dependsfileset`            Ist eine Dateiauswahl (`fileset`), die abhängige Dateien definiert? Diese werden bei der Prüfung, ob eingebunden Tasks ausgeführt werden sollen, mit berücksichtigt.

## 4.1.2 for

Der Task stellt eine Zählschleife zur Verfügung, die alle eingebetteten Tasks x-mal aufruft. Dabei wird die Schleife solange durchlaufen, wie die Zählvariable kleiner dem Endwert ist. Die verwendete Variable kann dabei bei den eingebetteten Tasks verwendet werden. Aber Achtung: Da die Variable global gehalten wird, kann bei Threads nicht gewährleistet werden, dass diese den korrekten Wert hat, wenn man auf diese über `#{i}` zugreift.

Jar-Archiv                      worker-base.jar

Parameter

**count**                      Legt die Schrittweite für die Schleife fest (Grundeinstellung 1).  
**end**                         Legt den Endwert für die Schleife fest.  
**param**                     Legt den Variablennamen für die Schleife fest (Grundeinstellung `i`). Die Variable wird nach der Schleife wieder gelöscht.  
**start**                      Legt den Startwert für die Schleife fest (Grundeinstellung 0).

Parameter geerbt von `de.nm.ant.parallel.AbstractParallelTask`

**exitonerror**                Legt fest, ob bei einem Fehler der Task abgebrochen werden soll (Grundeinstellung ja).  
**failonany**                 Legt fest, ob Tasks, die in der Warteschleife sind, abgebrochen werden sollen (Grundeinstellung nein).  
**keepalivetime**            Legt die Lebensdauer für einen warteten Task fest (Grundeinstellung 60s).  
**maxpoolsize**             Legt fest, wie viele Aufträge in die Warteschlange aufgenommen werden können.  
**threadcount**             Legt die Anzahl an gleichzeitigen Threads fest (Wird von 'threadsperprocessor' überschrieben, wenn dieser Parameter verwendet wird).  
**threadsperprocessor**     Legt die Anzahl der Threads für einen Prozessor fest. Das System ermittelt die Anzahl an Prozessoren und multipliziert dies mit der Anzahl an Threads pro Prozessors und legt somit die Anzahl der Threads fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                      Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).  
**deletescriptfile**        Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).  
**encoding**                 Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).  
**trim**                        Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).  
**usenewenvironment**     Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).  
**verbose**                    Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).  
**verboseheader**            Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

Innertasks geerbt von `de.nm.ant.parallel.AbstractParallelTask`

**dependsfileset**            Ist eine Dateiauswahl (fileset), die abhängige Dateien definiert? Diese werden bei der Prüfung, ob eingebunden Tasks ausgeführt werden sollen, mit berücksichtigt.

## 4.1.3 ifchecksum

Der Task ermittelt die Quelldateien und überprüft, ob die abhängigen Dateien älter oder nicht vorhanden sind. Ist dies der Fall, werden der Reihe nach alle eingebetteten Tasks aufgerufen. Vor dem Start der eingebetteten Tasks wird geprüft, ob folgende Parameter ('srcfile', 'destfile', 'workdir' und 'basename') übergeben werden können. Wenn ja, werden diese gesetzt. Wird 'threadcount' bzw. 'threadsperprocessor' gesetzt, so werden die Tasks entsprechend der Einstellung parallel über eine Queue ausgeführt.

Jar-Archiv                    worker-base.jar

Parameter

**checksum**                    Die Methode, welche die Prüfsumme ermittelt. Zur Zeit werden 'crc32', 'adler32', 'md5' und 'sha-256' unterstützt (Grundeinstellung: sha-256).

**prename**                    Das Kürzel, welches vor den Dateinamen (checksum.properties) gesetzt wird, so dass mehrere Propertiesdateien in einem Verzeichnis, unabhängig voneinander, verwendet werden können.

Parameter geerbt von `de.nm.ant.parallel.IfNewerTask`

**destdir**                    Das Zielverzeichnis.

**destextname**                Die Dateierweiterung zum Suchen für das Zielverzeichnis.

**srcdir**                    Das Quellverzeichnis.

**srcextname**                Die Dateierweiterung (regular expression) zum Suchen für das Quellverzeichnis.

Parameter geerbt von `de.nm.ant.parallel.AbstractParallelTask`

**exitonerror**                Legt fest, ob bei einem Fehler der Task abgebrochen werden soll (Grundeinstellung ja).

**failonany**                Legt fest, ob Tasks, die in der Warteschleife sind, abgebrochen werden sollen (Grundeinstellung nein).

**keepalivetime**            Legt die Lebensdauer für einen warteten Task fest (Grundeinstellung 60s).

**maxpoolsize**              Legt fest, wie viele Aufträge in die Warteschlange aufgenommen werden können.

**threadcount**              Legt die Anzahl an gleichzeitigen Threads fest (Wird von 'threadsperprocessor' überschrieben, wenn dieser Parameter verwendet wird.).

**threadsperprocessor**    Legt die Anzahl der Threads für einen Prozessor fest. Das System ermittelt die Anzahl an Prozessoren und multipliziert dies mit der Anzahl an Threads pro Prozessors und legt somit die Anzahl der Threads fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                    Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**        Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**                Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                    Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment**    Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**                Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**        Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

Innertasks

fileset                      Zusätzliche Auswahl an Sourcedateien über das fileset.

Innertasks geerbt von `de.nm.ant.parallel.IfNewerTask`

fileset                      Zusätzliche Auswahl an Quelldateien über das fileset.

Innertasks geerbt von `de.nm.ant.parallel.AbstractParallelTask`

dependsfileset              Ist eine Dateiauswahl (fileset), die abhängige Dateien definiert? Diese werden bei der Prüfung, ob eingebunden Tasks ausgeführt werden sollen, mit berücksichtigt.

## 4.1.4 ifdiff

Der Task ermittelt die Quelldateien und überprüft, ob die abhängigen Dateien unterschiedlich oder nicht vorhanden sind. Ist dies der Fall, wird die Datei vom Quellverzeichnis ins Zielverzeichnis kopiert und der Reihe nach alle eingebetteten Tasks aufgerufen. Vor dem Start der eingebetteten Tasks wird geprüft, ob folgende Parameter ('destdir', 'workdir', 'basename' und 'extname') übergeben werden können. Wenn ja, werden diese gesetzt. Wird 'threadcount' bzw. 'threadsperprocessor' gesetzt, so werden die Tasks entsprechend der Einstellung parallel über eine Queue ausgeführt.

Jar-Archiv                      worker-base.jar

Parameter

**destdir**                      Das Zielverzeichnis.  
**regexfile**                    Der reguläre Ausdruck (reg expr) zum Suchen für das Quellverzeichnis.  
**srcdir**                        Das Quellverzeichnis.

Parameter geerbt von `de.nm.ant.parallel.AbstractParallelTask`

**exitonerror**                Legt fest, ob bei einem Fehler der Task abgebrochen werden soll (Grundeinstellung ja).  
**failonany**                 Legt fest, ob Tasks, die in der Warteschleife sind, abgebrochen werden sollen (Grundeinstellung nein).  
**keepalivetime**            Legt die Lebensdauer für einen warteten Task fest (Grundeinstellung 60s).  
**maxpoolsize**             Legt fest, wie viele Aufträge in die Warteschlange aufgenommen werden können.  
**threadcount**             Legt die Anzahl an gleichzeitigen Threads fest (Wird von 'threadsperprocessor' überschrieben, wenn dieser Parameter verwendet wird.).  
**threadsperprocessor**    Legt die Anzahl der Threads für einen Prozessor fest. Das System ermittelt die Anzahl an Prozessoren und multipliziert dies mit der Anzahl an Threads pro Prozessors und legt somit die Anzahl der Threads fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                      Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).  
**deletescriptfile**        Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).  
**encoding**                 Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).  
**trim**                        Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).  
**usenewenvironment**    Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).  
**verbose**                    Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).  
**verboseheader**         Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

Innertasks

**fileset**                    Zusätzliche Auswahl an Quelldateien über das fileset.

Innertasks geerbt von `de.nm.ant.parallel.AbstractParallelTask`

**dependsfileset**         Ist eine Dateiauswahl (fileset), die abhängige Dateien definiert? Diese werden bei der Prüfung, ob eingebunden Tasks ausgeführt werden sollen, mit berücksichtigt.

## 4.1.5 ifnewer

Der Task ermittelt die Quelldateien und überprüft, ob die abhängigen Dateien älter oder nicht vorhanden sind. Ist dies der Fall, werden der Reihe nach alle eingebetteten Tasks aufgerufen. Vor dem Start der eingebetteten Tasks wird geprüft, ob folgende Parameter ('srcfile', 'destfile', 'workdir' und 'basename') übergeben werden können. Wenn ja, werden diese gesetzt. Wird 'threadcount' bzw. 'threadsperprocessor' gesetzt, so werden die Tasks entsprechend der Einstellung parallel über eine Queue ausgeführt.

Jar-Archiv worker-base.jar

Parameter

**destdir** Das Zielverzeichnis.  
**destextname** Die Dateierweiterung zum Suchen für das Zielverzeichnis.  
**srcdir** Das Quellverzeichnis.  
**srcextname** Die Dateierweiterung (regular expression) zum Suchen für das Quellverzeichnis.

Parameter geerbt von `de.nm.ant.parallel.AbstractParallelTask`

**exitonerror** Legt fest, ob bei einem Fehler der Task abgebrochen werden soll (Grundeinstellung ja).  
**failonany** Legt fest, ob Tasks, die in der Warteschleife sind, abgebrochen werden sollen (Grundeinstellung nein).  
**keepalivetime** Legt die Lebensdauer für einen warteten Task fest (Grundeinstellung 60s).  
**maxpoolsize** Legt fest, wie viele Aufträge in die Warteschlange aufgenommen werden können.  
**threadcount** Legt die Anzahl an gleichzeitigen Threads fest (Wird von 'threadsperprocessor' überschrieben, wenn dieser Parameter verwendet wird).  
**threadsperprocessor** Legt die Anzahl der Threads für einen Prozessor fest. Das System ermittelt die Anzahl an Prozessoren und multipliziert dies mit der Anzahl an Threads pro Prozessors und legt somit die Anzahl der Threads fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug** Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).  
**deletescriptfile** Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).  
**encoding** Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).  
**trim** Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).  
**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).  
**verbose** Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).  
**verboseheader** Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

Innertasks

**fileset** Zusätzliche Auswahl an Quelldateien über das fileset.

Innertasks geerbt von `de.nm.ant.parallel.AbstractParallelTask`

**dependsfileset** Ist eine Dateiauswahl (fileset), die abhängige Dateien definiert? Diese werden bei der Prüfung, ob eingebunden Tasks ausgeführt werden sollen, mit berücksichtigt.

## 4.2 Gruppendefinition

### 4.2.1 defgroup

Der Task definiert eine Gruppe von Tasks, die mit 'group' ausgeführt werden können. Somit lassen sich Tasks mehrmals in verschiedenen Bereichen verwenden, ohne dass diese erneut definiert werden müssen. Die definierte Gruppe kann in jedem Task eingebunden werden. Bei 'ifnewer' bzw. 'ifdiff' werden entsprechende Parameter gesetzt und an die eingebundenen Tasks weiter gegeben.

Jar-Archiv                 worker-base.jar

Parameter

**overwrite**                 Überschreibt schon definierte Gruppen (Grundeinstellung ja).  
**passthrough**             Legt fest, ob Parameter an eingebundene Tasks weitergegeben werden oder nicht (Grundeinstellung: ja).

Parameter geerbt von `de.nm.ant.group.AbstractGroup`

**name**                       Der Name der Gruppe.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                     Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**       Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**                 Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                      Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment**   Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**                 Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**         Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

## 4.2.2 group

Der Task dient dazu, gruppierte Tasks, die vorher mit 'defgroup' definiert worden sind, auszuführen. Vor der Ausführung wird geprüft, ob der Task folgende Parameter (bei 'ifnewer': 'workdir', 'srcfile', 'destfile' und 'basename' bzw. bei 'ifdiff': 'destdir', 'workdir', 'basename' und 'extname') aufnehmen kann. Wenn ja, werden diese Werte gesetzt.

Jar-Archiv                    worker-base.jar

Parameter

<b>basename</b>	Legt den Basisnamen fest.
<b>destdir</b>	Das Zielverzeichnis.
<b>destfile</b>	Die Zieldatei.
<b>extname</b>	Die Dateierweiterung.
<b>mode</b>	Legt fest, ob Parameter von 'ifdiff' oder 'ifnewer' übernommen worden sind.
<b>srcfile</b>	Die Quelldatei.
<b>workdir</b>	Das Arbeitsverzeichnis.

Parameter geerbt von `de.nm.ant.group.AbstractGroup`

<b>name</b>	Der Name der Gruppe.
-------------	----------------------

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

<b>debug</b>	Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).
<b>deletescriptfile</b>	Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).
<b>encoding</b>	Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).
<b>trim</b>	Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).
<b>usenewenvironment</b>	Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).
<b>verbose</b>	Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).
<b>verboseheader</b>	Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

## 4.3 Datei-Tools

### 4.3.1 copynamespace

Kopiert Dateien in einem Namespace in ein Zielverzeichnis. Dabei wird die Datei so umbenannt, dass der Namespace Teil des Dateinamens wird. Anstelle von Verzeichnistrennern wird ein Unterstrich verwendet.

Jar-Archiv                 worker-base.jar

Parameter

**replace**                 Variablen, die in der Datei mit `${...}` gesetzt sind, werden ersetzt (Grundeinstellung: false).

**todir**                   Legt das Verzeichnis fest, in das die Dateien kopiert werden.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                   Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**       Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**               Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                    Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment**     Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**                 Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**         Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

Innertasks

**fileset**                 Legt fest, welche Dateien kopiert werden sollen.

Beispiel(e):

```
<copynamespace todir="${target.dir}">
  <fileset dir="../worker-base/src/java">
    <include name="**/*.tex" />
  </fileset>
  <fileset dir="../worker-ipdf/src/java">
    <include name="**/*.tex" />
  </fileset>
  <fileset dir="../worker-tools/src/java">
    <include name="**/*.tex" />
  </fileset>
</copynamespace>
```

In dem Beispiel wird in den entsprechenden java-src-Verzeichnissen nach Dateien mit der Endung `.tex` gesucht. Diese werden dann in das Verzeichnis `target` kopiert. Dabei wird die Verzeichnisstruktur des Namensraums als Dateiname verwendet.

z. B. `de/nm/ant/ipdf/PdfInsertTask.tex` wird zu `de_nm_ant_ipdf_PdfInsertTask.tex`

## 4.3.2 copyreplace

Kopiert Dateien und ersetzt dabei alle Variablen, die mit `${...}` in der Datei definiert sind.

Jar-Archiv                 worker-base.jar

Parameter

**destdir**                 Das Zielverzeichnis.  
**srcdir**                 Das Quellverzeichnis, aus dem alle Unterverzeichnisse kopiert werden.  
**usesrcfile**             Diese Einstellung wird ignoriert und auf 'true' gesetzt.

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile**             Die Zielfile.  
**outname**             Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.  
**srcfile**             Die Quelldatei.  
**usesrcfile**           Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename**            Der Basisname einer Datei (ohne Erweiterung).  
**bufferize**            Legt die Puffergröße für das Dateihandling fest (Grundeinstellung 64kb).  
**newbasename**        Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.  
**workdir**             Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).  
**deletescriptfile**    Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).  
**encoding**            Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).  
**trim**                 Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).  
**usenewenvironment**  Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).  
**verbose**             Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).  
**verboseheader**      Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

### 4.3.3 copysubdir

Kopiert aus einem Verzeichnis alle Unterverzeichnisse in das Zielverzeichnis. Normale Dateien im Quellverzeichnis werden dabei nicht kopiert.

Jar-Archiv                 worker-base.jar

Parameter

**copydotfiles**           Legt fest, ob versteckte Dateien/Verzeichnisse (starten mit einem Punkt) kopiert werden sollen (Grundeinstellung nein).

**destdir**                 Das Zielverzeichnis.

**forceoverwrite**       Legt fest, ob Dateien überschrieben werden (Grundeinstellung nein).

**newer**                 Legt fest, ob Dateien nur kopiert werden, wenn diese neuer sind (Grundeinstellung ja).

**srcdir**                 Das Quellverzeichnis, aus dem alle Unterverzeichnisse kopiert werden.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                 Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**     Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**             Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                 Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment**   Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**             Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**       Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

## 4.3.4 removeother

Löscht Dateien mit gleichem Basisnamen und Erweiterungen, die nicht in der Liste enthalten sind.

Jar-Archiv                worker-base.jar

Parameter

**newbasename**            Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.

**othernames**             Legt die Liste (durch Komma getrennt) der Dateierweiterungen fest, die nicht gelöscht werden sollen ('tex,pdf').

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename**                Der Basisname einer Datei (ohne Erweiterung).

**buffersize**             Legt die Puffergröße für das Dateienhandling fest (Grundeinstellung 64kb).

**newbasename**            Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.

**workdir**                 Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                    Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**        Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**                Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                    Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment**     Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**                 Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**         Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

## 4.3.5 remove

Löscht die angegebene Datei.

Jar-Archiv                worker-base.jar

Parameter

**extname**                Legt die Dateierweiterung für die zu löschende Datei fest.

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename**                Der Basisname einer Datei (ohne Erweiterung).

**buffersize**              Legt die Puffergröße für das Dateihandling fest (Grundeinstellung 64kb).

**newbasename**            Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.

**workdir**                 Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                    Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**        Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**                Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                     Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment**     Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**                 Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**         Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

## 4.4 Konfiguration

### 4.4.1 configexecute

Legt fest, welches Kommando für einen speziellen Task unter einem Betriebssystem ausgeführt werden soll.

Jar-Archiv      worker-base.jar

Parameter

<b>deletescript</b>	Das erzeugte Skript wird nach der Ausführung (ohne Fehler) wieder gelöscht.
<b>name</b>	Legt fest, für welches Programm ein Kommando ausgeführt werden soll.
<b>osname</b>	Legt fest, für welches Betriebssystem das Kommando gilt (Ausdruck wird als RegEx verwendet).
<b>params</b>	Parameterliste mit einem Kommando und Parametern (durch Leerzeichen getrennt). Wird ein Leerzeichen als Parameter verwendet, so muss der Ausdruck in einfache Hochkommata gesetzt werden).
<b>scriptext</b>	Die Dateierweiterung für das Skript. Der Punkt muss mit angegeben werden.
<b>shellcmd</b>	Parameterliste mit einem Kommando für den Shellaufruf und Parametern (durch Leerzeichen getrennt). Wird ein Leerzeichen als Parameter verwendet, so muss der Ausdruck in einfache Hochkommata gesetzt werden).
Tasktext	Wird ein Text im Task verwendet, so wird ein Shell-Script mit den entsprechenden Text erstellt. Am Ende wird der erzeugte Befehl eingefügt.

Innertasks

**environment** Definiert für die Ausführung das Environment.

#### 4.4.2 configkv

Legt key-value-Parameter fest.

Jar-Archiv worker-base.jar

Parameter

**key**        Legt den Schlüssel (key) für den Eintrag fest.

**value**      Legt den Wert (value) für den Eintrag fest.

## 4.4.3 variable

Definiert und entfernt Variablen, ähnlich wie der Property-Task (dieser erzeugt nicht veränderbare Werte).

Jar-Archiv                 worker-base.jar

Parameter

**file**                     Lädt die Variablen aus einer properties Datei.  
**name**                    Der Variablenname.  
**osname**                 Legt fest, für welches Betriebssystem die Variable gesetzt wird (Ausdruck wird als RegEx verwendet). Wird der Wert nicht gesetzt, so wird die Variable immer gesetzt.  
**remove**                 Entferne die Variable wieder.  
**value**                   Der Wert.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                   Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).  
**deletescriptfile**       Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).  
**encoding**               Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).  
**trim**                    Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).  
**usenewenvironment**   Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).  
**verbose**                 Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).  
**verboseheader**         Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

## 4.4.4 executecmd

Definiert externe Kommandos für einzelne Programme.

Jar-Archiv                worker-base.jar

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**    Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**            Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**            Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**     Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

Innertasks

**configexecute**     Definiert die Konfiguration für die Ausführung.

## 4.5 PDF-Bearbeitung

### 4.5.1 pdfburst

Zerlegt eine PDF-Datei in einzelne Seiten.

Jar-Archiv                    worker-ipdf.jar

Parameter

**numberformat**            Legt das Format für die Nummer im Dateinamen fest.

Parameter geerbt von `de.nm.ant.ipdf.AbstractPagesPdfTask`

**pages**                    Legt die Seiten fest, die verwendet, bearbeitet, ... werden sollen (1,2,3,1-5,e,o,!5,\*).

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

**powner**                    Legt das owner-PW zum Öffnen fest.

**version**                    Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile**                    Die Zieldatei.

**outname**                    Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.

**srcfile**                    Die Quelldatei.

**usesrcfile**                Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename**                    Der Basisname einer Datei (ohne Erweiterung).

**bufferize**                    Legt die Puffergröße für das Dateienhandling fest (Grundeinstellung 64kb).

**newbasename**                Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.

**workdir**                    Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                    Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**        Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**                    Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                    Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment**        Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**                    Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**            Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

externe Jars                itext.jar bcmail-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

## 4.5.2 pdfcompress

Komprimiert den Inhalt einer PDF-Datei.

Jar-Archiv worker-ipdf.jar

Parameter

**compress** Legt den Kompressionsfaktor fest (0-9, full).

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

**powner** Legt das owner-PW zum Öffnen fest.

**version** Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile** Die Zieldatei.

**outname** Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.

**srcfile** Die Quelldatei.

**usesrcfile** Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename** Der Basisname einer Datei (ohne Erweiterung).

**bufferize** Legt die Puffergröße für das Dateienhandling fest (Grundeinstellung 64kb).

**newbasename** Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.

**workdir** Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug** Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile** Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding** Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim** Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose** Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader** Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

externe Jars itext.jar bemail-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

## 4.5.3 pdfcroppage

Beschneidet die Seiten eines PDF-Dokuments.

Jar-Archiv worker-ipdf.jar

Parameter

**addOnePoint** Bei 'autocrop' wird llx, lly, urx und ury auf 1 gesetzt.

**autocrop** Es wird der Rand abgeschnitten. Dazu wird das Programm 'gs' aufgerufen, um die BoundingBox zu ermitteln. llx, lly, urx und ury werden dabei zu den ermittelten Werten hinzugefügt.

**llx** Wert für BoundingBox

**lly** Wert für BoundingBox

**urx** Wert für BoundingBox

**ury** Wert für BoundingBox

**useHiRes** Es wird 'HiResBoundingBox' anstelle von 'BoundingBox' (gs-Aufruf) verwendet.

Parameter geerbt von `de.nm.ant.ipdf.AbstractPagesPdfTask`

**pages** Legt die Seiten fest, die verwendet, bearbeitet, ... werden sollen (1,2,3,1-5,e,o,!5,\*).

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

**powner** Legt das owner-PW zum Öffnen fest.

**version** Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile** Die Zieldatei.

**outname** Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.

**srcfile** Die Quelldatei.

**usesrcfile** Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename** Der Basisname einer Datei (ohne Erweiterung).

**bufferize** Legt die Puffergröße für das Dateienhandling fest (Grundeinstellung 64kb).

**newbasename** Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.

**workdir** Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug** Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile** Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding** Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim** Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose** Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

<b>verboseheader</b>	Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).
externe Programme	gs
externe Jars	itext.jar bcpmail-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

## 4.5.4 pdfinsert

Bindet PDF-Seiten (leere oder aus einer Datei) ein.

Jar-Archiv                worker-ipdf.jar

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

**owner**                Legt das owner-PW zum Öffnen fest.

**version**              Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile**             Die Zieldatei.

**outname**             Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.

**srcfile**              Die Quelldatei.

**usesrcfile**          Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename**            Der Basisname einer Datei (ohne Erweiterung).

**bufferize**            Legt die Puffergröße für das Dateihandling fest (Grundeinstellung 64kb).

**newbasename**        Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.

**workdir**             Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**    Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**            Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**            Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**     Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

Innertasks

**pages**                Legt fest, wo leere Seiten eingebunden werden sollen.

**pdffilepages**        Legt fest, welche Seiten eines PDFs wo eingebunden werden sollen.

externe Jars            itext.jar bcmail-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

Beispiel(e):

```
<pdfinsert srcfile="src/pdfs/onjcontrol.pdf"
  workdir="target"
  outname="$basename-insert1">
  <pdffilepages file="src/pdfs/s4.pdf" beforepage="1" pages="2-3" />
  <pdffilepages file="src/pdfs/i2.pdf" beforepage="5" />
  <pdffilepages file="src/pdfs/s4.pdf" beforepage="-1" pages="*" />
  <pdffilepages file="src/pdfs/ie.pdf" afterpage="-1" pages="1" />
</pdfinsert>
```

Das Beispiel erzeugt eine neue PDF-Datei im Verzeichnis `target` mit dem Namen `onjcontrol-insert1.pdf`. Folgende Seiten werden eingefügt.

- Aus der Datei `s4.pdf` werden die Seite 2-3 vor der Seite 1 eingefügt.
- Aus der Datei `i2.pdf` werden alle Seiten vor der Seite 5 eingefügt.
- Aus der Datei `s4.pdf` wird die letzte Seite vor der letzten Seite eingefügt.
- Aus der Datei `ie.pdf` wird die Seite 1 nach der letzten Seite eingefügt.

## 4.5.5 pdfmerge

Bindet PDF-Seiten (leere oder aus einer Datei) in ein neu erzeugtes Dokument ein.

Jar-Archiv worker-ipdf.jar

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

**pwner** Legt das owner-PW zum Öffnen fest.  
**version** Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile** Die Zielfile.  
**outname** Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.  
**srcfile** Die Quelldatei.  
**usesrcfile** Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename** Der Basisname einer Datei (ohne Erweiterung).  
**bufferize** Legt die Puffergröße für das Dateienhandling fest (Grundeinstellung 64kb).  
**newbasename** Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.  
**workdir** Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug** Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).  
**deletescriptfile** Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).  
**encoding** Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).  
**trim** Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).  
**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).  
**verbose** Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).  
**verboseheader** Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

Innertasks

**pages** Legt fest, wo leere Seiten eingebunden werden sollen.  
**pdffilepages** Legt fest, welche Seiten eines PDFs wo eingebunden werden sollen.  
**externe Jars** itext.jar bcmail-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

Beispiel(e):

```
<pdfmerge workdir="${target.dir}" basename="hugo" >
  <pdffilepages file="src/pdfs/s4.pdf" pages="2-3" />
  <pdffilepages file="src/pdfs/i2.pdf" />
  <pdffilepages file="src/pdfs/s4.pdf" pages="*" />
  <pdffilepages file="src/pdfs/ie.pdf" pages="1" />
</pdfmerge>
```

Das Beispiel erzeugt eine neue PDF-Datei im Verzeichnis `target` mit dem Namen `hugo.pdf`. Folgende Seiten werden eingefügt.

- Aus der Datei `s4.pdf` werden die Seite 2-3 eingefügt.
- Aus der Datei `i2.pdf` werden alle Seiten eingefügt.
- Aus der Datei `s4.pdf` wird die letzte Seite eingefügt.
- Aus der Datei `ie.pdf` wird die Seite 1 eingefügt.

## 4.5.6 pdfmetadata

Legt Metadaten für ein PDF-Dokument fest.

Jar-Archiv worker-ipdf.jar

Parameter

**author** Legt den Autor fest.  
**creator** Legt den Erzeuger fest.  
**keywords** Legt die Schlüsselwörter fest.  
**subject** Legt die Beschreibung fest.  
**title** Legt den Titel fest.

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

**pwner** Legt das owner-PW zum Öffnen fest.  
**version** Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile** Die Zieldatei.  
**outname** Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.  
**srcfile** Die Quelldatei.  
**usesrcfile** Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename** Der Basisname einer Datei (ohne Erweiterung).  
**bufferize** Legt die Puffergröße für das Dateienhandling fest (Grundeinstellung 64kb).  
**newbasename** Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.  
**workdir** Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug** Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).  
**deletescriptfile** Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).  
**encoding** Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).  
**trim** Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).  
**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).  
**verbose** Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).  
**verboseheader** Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

externe Jars itext.jar bcmail-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

## 4.5.7 pdfnup

Setzt mehrere PDF-Seiten auf einer Seite.

Jar-Archiv                worker-ipdf.jar

Parameter

**center**                Die Seiten werden im virtuellen Rahmen zentriert.

**hgap**                 Legt den horizontalen Zwischenraum fest.

**hpages**              Legt die Anzahl der horizontalen Seiten fest.

**landscape**          Es wird Querformat verwendet.

**margin**              Legt den Rand für gerade/ungerade Seiten fest.

**margineven**        Legt den Rand für gerade Seiten fest.

**marginodd**         Legt den Rand für ungerade Seiten fest.

**pagesize**           Legt die Papiergröße fest.

**vgap**                Legt den vertikalen Zwischenraum fest.

**vpages**             Legt die Anzahl der vertikalen Seiten fest.

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

**powner**             Legt das owner-PW zum Öffnen fest.

**version**            Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile**            Die Zieldatei.

**outname**            Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.

**srcfile**             Die Quelldatei.

**usesrcfile**        Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename**          Der Basisname einer Datei (ohne Erweiterung).

**bufferize**          Legt die Puffergröße für das Dateienhandling fest (Grundeinstellung 64kb).

**newbasename**      Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.

**workdir**            Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**              Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**  Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**          Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**            Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**    Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

externe Jars            itext.jar bcmal-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

## 4.5.8 pdfremovinglaunchaction

Entfernt 'launchaction' aus dem PDF-Dokument.

Jar-Archiv worker-ipdf.jar

Parameter

**javascript** Der JavaScript-Code, der anstelle der Aktion ausgeführt werden soll.

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

**password** Legt das owner-PW zum Öffnen fest.

**version** Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile** Die Zielfile.

**outname** Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.

**srcfile** Die Quelldatei.

**usesrcfile** Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename** Der Basisname einer Datei (ohne Erweiterung).

**bufferize** Legt die Puffergröße für das Dateihandling fest (Grundeinstellung 64kb).

**newbasename** Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.

**workdir** Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug** Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile** Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding** Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim** Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose** Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader** Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

externe Jars itext.jar bcpmail-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

## 4.5.9 pdfrotate

Dreht einzelne Seiten eines PDF-Dokuments.

Jar-Archiv worker-ipdf.jar

Parameter

**angle** Der Winkel.

Parameter geerbt von `de.nm.ant.ipdf.AbstractPagesPdfTask`

**pages** Legt die Seiten fest, die verwendet, bearbeitet, ... werden sollen (1,2,3,1-5,e,o,!5,\*).

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

**pwner** Legt das owner-PW zum Öffnen fest.

**version** Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile** Die Zieldatei.

**outname** Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.

**srcfile** Die Quelldatei.

**userfile** Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename** Der Basisname einer Datei (ohne Erweiterung).

**bufferize** Legt die Puffergröße für das Dateienhandling fest (Grundeinstellung 64kb).

**newbasename** Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.

**workdir** Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug** Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile** Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding** Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim** Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose** Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader** Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

externe Jars itext.jar bcpmail-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

## 4.5.10 pdfsecure

Setzt die Rechte eines PDF-Dokuments unter Verwendung eines 'owner'- und 'user'-PWs.

Jar-Archiv worker-ipdf.jar

Parameter

<b>aes</b>	Verwendet AES-Verschlüsselung.
<b>allow_assembly</b>	Erlaubt Inhalte zu extrahieren.
<b>allow_copy</b>	Erlaubt Kopieren, zum Beispiel in die Zwischenablage.
<b>allow_degraded_printing</b>	Erlaubt Drucken mit niedriger Auflösung.
<b>allow_file_in</b>	Erlaubt das Extrahieren eingebundener Dateien.
<b>allow_modify_annotations</b>	Erlaubt das Ändern von Annotations.
<b>allow_printing</b>	Erlaubt das Drucken.
<b>allow_screenreaders</b>	Erlaubt das Benutzen von Screenreadern.
<b>do_not_encrypt_metadata</b>	Verhindert das Verschlüsseln von Metadaten.
<b>newpowner</b>	Legt das neue Owner-PW fest.
<b>newpwuser</b>	Legt das neue User-PW fest.

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

<b>pwowner</b>	Legt das owner-PW zum Öffnen fest.
<b>version</b>	Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

<b>destfile</b>	Die Zieldatei.
<b>outname</b>	Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.
<b>srcfile</b>	Die Quelldatei.
<b>usesrcfile</b>	Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

<b>basename</b>	Der Basisname einer Datei (ohne Erweiterung).
<b>bufferize</b>	Legt die Puffergröße für das Dateihandling fest (Grundeinstellung 64kb).
<b>newbasename</b>	Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.
<b>workdir</b>	Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

<b>debug</b>	Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).
<b>deletescriptfile</b>	Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).
<b>encoding</b>	Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).
<b>trim</b>	Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).
<b>usenewenvironment</b>	Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).
<b>verbose</b>	Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

<b>verboseheader</b>	Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quell-datei ausgegeben werden soll (Grundeinstellung ja).
externe Jars	itext.jar bcpmail-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

## 4.5.11 pdfwatermark

Fügt Wasserzeichen in ein PDF-Dokument ein.

Jar-Archiv worker-ipdf.jar

Parameter

**angle** Legt den Winkel für den Text/Bild fest.  
**fontsize** Legt die Fontgröße für den Text fest.  
**image** Legt das Hintergrundbild fest (hat Vorrang vor dem Text!).  
**over** Das Wasserzeichen wird über den Inhalt oder hinter den Inhalt gezeichnet.  
**text** Legt den Text fest.

Parameter geerbt von `de.nm.ant.ipdf.AbstractPagesPdfTask`

**pages** Legt die Seiten fest, die verwendet, bearbeitet, ... werden sollen (1,2,3,1-5,e,o,!5,\*).

Parameter geerbt von `de.nm.ant.ipdf.AbstractPdfTask`

**powner** Legt das owner-PW zum Öffnen fest.  
**version** Legt die zu verwendende PDF-Version fest (Grundeinstellung: 5)

Parameter geerbt von `de.nm.ant.AbstractSrcDestTask`

**destfile** Die Zieldatei.  
**outname** Legt den Namen für die Ausgabedatei fest. Wird darin '\$basename' verwendet, so wird dieser mit dem aktuellen 'basename' ersetzt.  
**srcfile** Die Quelldatei.  
**usesrcfile** Legt fest, ob zuerst die Einstellung 'srcfile' verwendet werden soll oder 'workdir' + 'basename' + 'extname' (Grundeinstellung ja).

Parameter geerbt von `de.nm.ant.AbstractWorkdirTask`

**basename** Der Basisname einer Datei (ohne Erweiterung).  
**bufferize** Legt die Puffergröße für das Dateienhandling fest (Grundeinstellung 64kb).  
**newbasename** Der neu zu verwendende Basisname. Enthält dieser '\$basename', so wird der ursprüngliche Basisname eingesetzt.  
**workdir** Legt das Arbeitsverzeichnis fest.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug** Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).  
**deletescriptfile** Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).  
**encoding** Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).  
**trim** Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).  
**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).  
**verbose** Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).  
**verboseheader** Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).  
 externe Jars itext.jar bcmail-jdk16-145.jar bcprov-jdk16-145.jar bctsp-jdk16-145.jar

### 4.5.12 pages

Legt fest, wo leere Seiten in ein PDF-Dokument eingebunden werden.

Jar-Archiv worker-ipdf.jar

Parameter

**afterpage** Legt die Seite fest, nach der eine Seite eingefügt worden ist.

**beforepage** Legt die Seite fest, vor der eine Seite eingefügt worden ist.

**landscape** Die Seite wird im Querformat eingebunden.

**pagesize** Die Größe der Seite (Grundeinstellung: A4).

externe Jars itext.jar

Beispiel(e):

```
<pdfinsert srcfile="src/pdfs/onjcontrol.pdf"
  workdir="target"
  outname="$basename-insert2">
  <pages beforepage="1" pagesize="A4" landscape="true" />
  <pages afterpage="-1" pagesize="A4" landscape="true" />
  <pages afterpage="2" pagesize="A4" landscape="true" />
</pdfinsert>
```

Das Beispiel erzeugt eine neue PDF-Datei im Verzeichnis target mit dem Namen onjcontrol-insert2.pdf. Folgende leere Seiten werden eingefügt.

- Eine leere Seite (A4-Quer) vor der ersten Seite.
- Eine leere Seite (A4-Quer) nach der letzten Seite.
- Eine leere Seite (A4-Quer) nach der zweiten Seite.

## 4.5.13 pdffilepages

Legt fest, welche PDF-Seiten aus einer Datei in ein PDF-Dokument eingebunden werden.

Jar-Archiv worker-ipdf.jar

Parameter

**file** Legt die PDF-Datei fest, von der Seiten eingebunden werden sollen.

**landscape** Wird ignoriert.

**pages** Legt die Seiten fest, die eingebunden werden sollen.

**pagesize** Wird ignoriert.

Parameter geerbt von `de.nm.ant.ipdf.types.Pages`

**afterpage** Legt die Seite fest, nach der eine Seite eingefügt worden ist.

**beforepage** Legt die Seite fest, vor der eine Seite eingefügt worden ist.

**landscape** Die Seite wird im Querformat eingebunden.

**pagesize** Die Größe der Seite (Grundeinstellung: A4).

externe Jars itext.jar

Beispiel(e):

```
<pdfinsert srcfile="src/pdfs/onjcontrol.pdf"
  workdir="target"
  outname="$basename-insert1">
  <pdffilepages file="src/pdfs/s4.pdf" beforepage="1" pages="2-3" />
  <pdffilepages file="src/pdfs/i2.pdf" beforepage="5" />
  <pdffilepages file="src/pdfs/s4.pdf" beforepage="-1" pages="*" />
  <pdffilepages file="src/pdfs/ie.pdf" afterpage="-1" pages="1" />
</pdfinsert>
```

Das Beispiel erzeugt eine neue PDF-Datei im Verzeichnis `target` mit dem Namen `onjcontrol-insert1.pdf`. Folgende Seiten werden eingefügt.

- Aus der Datei `s4.pdf` werden die Seiten 2-3 vor der Seite 1 eingefügt.
- Aus der Datei `i2.pdf` werden alle Seiten vor der Seite 5 eingefügt.
- Aus der Datei `s4.pdf` wird die letzte Seite vor der letzten Seite eingefügt.
- Aus der Datei `ie.pdf` wird die Seite 1 nach der letzten Seite eingefügt.

## 4.6 Utility

### 4.6.1 systeminfo

Gibt die Systeminformationen, die in den System-Properties gespeichert sind, aus.

Jar-Archiv                worker-base.jar

Parameter

**filter**                Der Filter (regular Expression) für die Properties.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**    Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**            Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment** Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**            Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**     Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

## 4.6.2 timer

Ermittelt die Laufzeit von eingebetteten Tasks mit Hilfe von `System.currentTimeMillis()`.

Jar-Archiv                      worker-base.jar

Parameter

**name**                              Der Name des Timers.

Parameter geerbt von `de.nm.ant.AbstractWorkerTask`

**debug**                              Legt fest, ob debug Informationen ausgegeben werden sollen (Grundeinstellung nein).

**deletescriptfile**      Löscht eine evtl. Skriptdatei nach der Ausführung, wenn kein Fehler aufgetreten ist (Grundeinstellung ja).

**encoding**                          Legt das Encoding für erzeugte Textdateien fest (Grundeinstellung UTF-8).

**trim**                                  Wird Text hinzugefügt, werden führende und nachfolgende Leerzeichen entfernt (Grundeinstellung ja).

**usenewenvironment**      Legt fest, ob für externe Aufruf ein neues Environment verwendet werden soll (Grundeinstellung nein).

**verbose**                              Legt fest, ob Meldungen oder Programmausgaben angezeigt werden sollen (Grundeinstellung nein).

**verboseheader**              Legt fest, ob eine Anfangsmeldung mit dem Tasknamen und der Quelldatei ausgegeben werden soll (Grundeinstellung ja).

## 5 Nutzungsrechte, Lizenz und Sourcecode

Durch die verschiedenen Nutzungsrechte und Lizenzen muss leider jedes einzelne Paket unter Umständen mit einem anderen Nutzungsrecht bzw. einer Lizenz ausgestattet werden.

### 5.1 Apache-ANT

Apache-ANT steht unter der „Apache License 2.0“

Die Apache-Lizenz ist die Freie-Software-Lizenz der Apache Software Foundation. Die aktuelle Version 2.0 wurde im Januar 2004 veröffentlicht. Sie wurde gegenüber der vorherigen Version 1.1 stark erweitert. Aufgrund ihres Umfangs wird in den Quelltexten der einzelnen Apache-Projekte nicht mehr der komplette Text, sondern lediglich ein Verweis auf die Originallizenz eingefügt. Die Apache-Lizenz wird von der Free Software Foundation als Lizenz für freie Software anerkannt und ist zu der GNU General Public License Version 3, nicht jedoch zur Version 2, kompatibel.

Prinzipiell beinhaltet sie:

- Man darf Software unter dieser Lizenz frei in jedem Umfeld verwenden, modifizieren und verteilen.
- Wenn man sie verteilt, muss eindeutig darauf hingewiesen werden, welche Software unter der Apache-Lizenz verwendet wurde und dass diese vom Lizenzgeber (name of copyright owner) stammt. Eine Kopie der Lizenz muss dem Paket beiliegen.
- Änderungen am Quellcode der unter der Apache-Lizenz stehenden Software brauchen nicht zum Lizenzgeber zurückgeschickt zu werden.
- Eigene Software, die unter Apache-Lizenz stehende Software verwendet, braucht **nicht** unter der Apache-Lizenz zu stehen.
- Die eigene Software darf nur dann Apache heißen, wenn eine schriftliche Genehmigung der Apache Foundation vorliegt.

Quelle: <https://secure.wikimedia.org/wikipedia/de/wiki/Apache-Lizenz>

## 5.2 Worker-Base

Das Paket Worker-Base ist Freeware.

Diese Software ist Freeware:

- Die Software darf frei verwendet werden, privat und kommerziell.
- Eine Verbreitung gegen ein Entgelt etc. ist untersagt.
- Bei Verbreitung darf das Paket nur komplett, mit allen Teilen (z. B. auch Dokumentation), weitergegeben werden.
- Es wird keinerlei Haftung für die Software und deren Einsatz übernommen.

DIESE SOFTWARE WIRD VOM AUTOR OHNE JEGLICHE SPEZIELLE ODER IMPLIZIERTE GARANTIE ZUR VERFÜGUNG GESTELLT, DIE UNTER ANDEREM EINSCHLIESSEN: DIE IMPLIZIERTE GARANTIE DER VERWENDBARKEIT DER SOFTWARE FÜR EINEN BESTIMMTEN ZWECK. AUF KEINEN FALL IST DER AUTOR FÜR IRGENDWELCHE DIREKTEN, INDIREKTEN, ZUFÄLLIGEN, SPEZIELLEN, BEISPIELHAFTEN ODER FOLGESCHÄDEN (UNTER ANDEREM VERSCHAFFEN VON ERSATZGÜTERN ODER -DIENSTLEISTUNGEN; EINSCHRÄNKUNG DER NUTZUNGSFÄHIGKEIT; VERLUST VON NUTZUNGSFÄHIGKEIT; DATEN; PROFIT ODER GESCHÄFTSUNTERBRECHUNG), WIE AUCH IMMER VERURSACHT UND UNTER WELCHER VERPFLICHTUNG AUCH IMMER, OB IN VERTRAG, STRIKTER VERPFLICHTUNG ODER UNERLAUBTER HANDLUNG (INKLUSIVE FAHRLÄSSIGKEIT) VERANTWORTLICH, AUF WELCHEM WEG SIE AUCH IMMER DURCH DIE BENUTZUNG DIESER SOFTWARE ENTSTANDEN SIND, SOGAR, WENN SIE AUF DIE MÖGLICHKEIT EINES SOLCHEN SCHADENS HINGEWIESEN WORDEN SIND.

## 5.3 Worker-ipdf

In Absprache mit dem iText-Autor *Bruno Lowagie* steht das Paket Worker-ipdf, wie iText selbst, unter der GNU Affero General Public License version 3.

Sourcecode

[PdfRotateTask.java](#)

[InsertPageRule.java](#)

[PageRule.java](#)

[PageRange.java](#)

[Pages.java](#)

[PdfFilePages.java](#)

[PdfRemovingLaunchActionTask.java](#)

[PdfCropPageTask.java](#)

[PdfCompressTask.java](#)

[AbstractPdfTask.java](#)

[PdfInsertTask.java](#)

[PdfNupTask.java](#)

[PdfMetaDataTask.java](#)

[AbstractPagesPdfTask.java](#)

[PdfMergeTask.java](#)

[PdfWatermarkTask.java](#)

[PdfSecureTask.java](#)

[PdfBurstTask.java](#)

## 5.4 Worker-Tools

Das Paket Worker-Tools ist Freeware.

Diese Software ist Freeware:

- Die Software darf frei verwendet werden, privat und kommerziell.
- Eine Verbreitung gegen ein Entgelt etc. ist untersagt.
- Bei Verbreitung darf das Paket nur komplett, mit allen Teilen (z. B. auch Dokumentation), weitergegeben werden.
- Es wird keinerlei Haftung für die Software und deren Einsatz übernommen.

DIESE SOFTWARE WIRD VOM AUTOR OHNE JEGICHE SPEZIELLE ODER IMPLIZIERTE GARANTIE ZUR VERFÜGUNG GESTELLT, DIE UNTER ANDEREM EINSCHLIESSEN: DIE IMPLIZIERTE GARANTIE DER VERWENDBARKEIT DER SOFTWARE FÜR EINEN BESTIMMTEN ZWECK. AUF KEINEN FALL IST DER AUTOR FÜR IRGENDWELCHE DIREKTEN, INDIREKTEN, ZUFÄLLIGEN, SPEZIELLEN, BEISPIELHAFTEN ODER FOLGESCHÄDEN (UNTER ANDEREM VERSCHAFFEN VON ERSATZGÜTERN ODER -DIENSTLEISTUNGEN; EINSCHRÄNKUNG DER NUTZUNGSFÄHIGKEIT; VERLUST VON NUTZUNGSFÄHIGKEIT; DATEN; PROFIT ODER GESCHÄFTSUNTERBRECHUNG), WIE AUCH IMMER VERURSACHT UND UNTER WELCHER VERPFLICHTUNG AUCH IMMER, OB IN VERTRAG, STRIKTER VERPFLICHTUNG ODER UNERLAUBTER HANDLUNG (INKLUSIVE FAHRLÄSSIGKEIT) VERANTWORTLICH, AUF WELCHEM WEG SIE AUCH IMMER DURCH DIE BENUTZUNG DIESER SOFTWARE ENTSTANDEN SIND, SOGAR, WENN SIE AUF DIE MÖGLICHKEIT EINES SOLCHEN SCHADENS HINGEWIESEN WORDEN SIND.

## 6 ToDo

- Dokumentation auf Rechtschreibung und Sprache prüfen
- FAQ erstellen
- weitere build.xml Beispiele erzeugen
- Wer hat mitgeholfen?
- Nutzungsrechte / Lizenzen überprüfen
- ...

## Übersicht Ant-Task

<b>C</b>		<b>P</b>	
clean .....	6	pages .....	49
configexecute .....	14, 30	parallelqueue .....	8
configkv .....	31	pdfburst .....	34
copynamespace .....	25	pdfcompress .....	35
copyreplace .....	26	pdfcropage .....	36
copysubdir .....	27	pdfcropage .....	36
		pdfinsert .....	50
<b>D</b>		pdfinsert .....	38
d2t2pdf .....	14	pdfmerge .....	40
defgroup .....	8, 23	pdfmetadata .....	42
		pdfnup .....	43
<b>E</b>		pdfremovinglaunchaction .....	44
executecmd .....	13, 33	pdfrotate .....	45
		pdfsecure .....	5, 6, 46
		pdfwatermark .....	48
<b>F</b>		<b>R</b>	
for .....	18	remove .....	29
foreach .....	16	removeother .....	28
<b>G</b>		<b>S</b>	
group .....	8, 24	systeminfo .....	13, 51
<b>I</b>		<b>T</b>	
ifchecksum .....	8, 19	timer .....	52
ifdiff .....	8, 21	<b>V</b>	
ifnewer .....	6, 8, 22	variable .....	32
init .....	6		

## Übersicht externe Programme

G	
gs .....	<a href="#">37</a>

## Übersicht JAR-Archive

### B

bcmal-jdk16-145.jar	5, 34, 35, 37, 38, 40, 42–45, 47, 48
bcprov-jdk16-145.jar	5, 34, 35, 37, 38, 40, 42–45, 47, 48
bctsp-jdk16-145.jar ..	5, 34, 35, 37, 38, 40, 42–45, 47, 48

### I

itext.jar .....	5, 34, 35, 37, 38, 40, 42–45, 47–50
-----------------	-------------------------------------

### W

worker-base.jar .....	5
Task: configexecute .....	30
Task: configkv .....	31
Task: copynamespace .....	25
Task: copyreplace .....	26
Task: copysubdir .....	27
Task: defgroup .....	23
Task: executecmd .....	33
Task: for .....	18
Task: foreach .....	16
Task: group .....	24
Task: ifchecksum .....	19
Task: ifdiff .....	21

Task: ifnewer .....	22
Task: remove .....	29
Task: removeother .....	28
Task: systeminfo .....	51
Task: timer .....	52
Task: variable .....	32
worker-ipdf.jar .....	5
Task: pages .....	49
Task: pdfburst .....	34
Task: pdfcompress .....	35
Task: pdfcroppage .....	36
Task: pdfcropimage .....	50
Task: pdfinsert .....	38
Task: pdfmerge .....	40
Task: pdfmetadata .....	42
Task: pdfnup .....	43
Task: pdfremovinglaunchaction .....	44
Task: pdfrotate .....	45
Task: pdfsecure .....	46
Task: pdfwatermark .....	48
worker-tools.jar .....	5

## Übersicht Klassen / Interfaces

<b>A</b>		<b>G</b>	
AbstractPagesPdfTask .....	11	GroupTask .....	8
AbstractParallelTask .....	8	<b>I</b>	
AbstractPdfTask .....	11	IfChecksumTask .....	8
AbstractSrcDestTask .....	10	IfDiffTask .....	8
AbstractTeXTask .....	12	IfNewerTask .....	8
AbstractTeXToolTask .....	12	<b>P</b>	
AbstractWorkdirTask .....	9	ParallelQueueTask .....	8
AbstractWorkerTask .....	7	<b>T</b>	
<b>D</b>		Task .....	7
DefGroupTask .....	8		

# Stichwortverzeichnis

<b>A</b>			
adler32 .....	8	md5 .....	8
Apache-ANT .....	4	<b>N</b>	
Aufbau .....	4	Nutzungsrecht .....	53
<b>B</b>		<b>P</b>	
bash .....	13	PATH .....	13
Basisfunktionalität .....	5	PDF	
Bibliothek .....	5	Bearbeitung .....	5
einbinden .....	5	Verschlüsselung .....	5
<b>C</b>		<b>S</b>	
CLASSPATH .....	5	sha-256 .....	8
CPU-Kern .....	6	Sourcecode .....	53
crc32 .....	8	Systemproperties .....	13
<b>E</b>		<b>T</b>	
Erweiterung .....	4, 5	Thread .....	5
export .....	13	<b>U</b>	
<b>L</b>		Umgebungsvariable .....	5, 13
Lizenz .....	4, 53	Unterstützerklassen .....	5
<b>M</b>		<b>W</b>	
make .....	4	Worker .....	4