

Dokumentation zum Contenido Plugin Manager Version 1.0.0

Stand: 20. März 2008

Autor: Frederic Schneider (4fb)

XML-Schema plugin_info.xml

Die XML-Datei definiert, wie das Plugin heißt und was während der Installation passieren soll. Werte, die nicht blau markiert sind, sind optional und besitzen einen Standardwert oder sind, sofern nicht zwingend für den Betrieb notwendig, leer.

General-Tag

In diesem Bereich werden die Grundeinstellungen des Plugins definiert.

1. *plugin_name*: Definiert den Ausgabenamen des Plugins
2. *plugin_foldername*: Definiert den Namen des Ordners unterhalb von Contenido-Plugin-Ordners
3. *guid*: Von der Four For Business AG vergebene ID zur Identifikation des Plugins
4. *description*: Kurzbeschreibung, was das Plugin macht, in englischer Sprache
5. *author*: Name des Plugin-Autors bzw. der Entwicklergruppe
6. *copyright*: Copyright des Plugins
7. *version*: Versionsnummer des vorliegenden Plugins
8. *dependencies*: Definiert die Abhängigkeiten zu anderen Plugins, durch Semikolon getrennt
9. *min_contenido_version*: Welche Contenido-Version setzt das Plugin voraus?
Standardwert: 5.0.0

Dependencies-Tag

```
<dependencies>
  <depend version="1.0.3">b9653 ztdbde004
</dependencies>
```

Optional können Abhängigkeiten zu anderen Plugins vorausgesetzt werden. Der Plugin Manager erkennt Dritte Plugins anhand der „guid“. Der „dependencies“-Tag erwartet öffnende und schließende „depend“-Tags sowie die Eigenschaft „version“, die dem „version“-Tag des „general“-Bereiches Dritter Plugins entspricht.

Settings-Tag

Definiert, was während der Installation passieren soll

```
<active>1</active>
<sql>1</sql>
```

1. *active*: Der Wert 1 steht für aktiv, der Wert 0 für inaktiv; Standard: 1
2. *sql*: Der Plugin Manager liest automatisch (Standard-Wert: 1) alle SQL-Dateien aus dem Ordner „sql“ heraus und führt diese aus. Dies kann deaktiviert werden, indem der Wert auf 0 gesetzt wird

Beide Tags können weggelassen werden, sofern das Plugin aktiviert sein und alle SQL-Dateien auslesen soll.

```

<areas>
  <area>linkchecker</area>
  <area parent="linkchecker">lc wh:

```

3. *areas*: Innerhalb des „areas“-Tag können beliebig viele Areas definiert werden, die in die Datenbank geschrieben werden. Eine Area wird über den öffnenden und schließenden „area“-Tag definiert wird. Optional besitzt der „area“-Tag die Eigenschaft „parent“. Diese ist dazu nützlich, eine Area einer anderen unter zu ordnen. Es ist der jeweilige Area-Name anzugeben.

```

<actions>
  <action area="linkchecker">linkch
</actions>

```

4. *actions*: Es können beliebig viele Actions definiert werden, indem der öffnende und schließende „action“-Tag geschrieben wird. Er besitzt die notwendige Eigenschaft „area“, die den Namen einer passenden Area erwartet.

```

<frames>
  <frame area="linkchecker" folder="includes" name="include.linkchecker.php" frame="4" />
  <frame area="linkchecker" folder="includes" name="include.linkchecker_right_top.php" frame="3" />
  <frame area="lc_whitelist" folder="includes" name="include.linkchecker_whitelist.php" frame="3" />
</frames>

```

5. *frames*: Den verschiedenen Dateien aus dem „includes“-Ordner können Frames zugewiesen werden. Dafür besitzt der „frames“-Tag die beliebig oft einzusetzenden „frame“-Tags, die die Eigenschaften „area“ (Name der passenden Area), „folder“ (Name des Ordners, indem sich die passende Datei befindet, z. B. „includes“ oder „includes/cache“) sowie „name“ (Name der Datei, z. B. „include.plugin_steps.php“) haben. Der Plugin Manager schreibt automatisch Einträge in der Contendio-„Files“-Tabelle, die mit der Contendio-„Frames“-Tabelle verknüpft werden.

```

<nav_sub>
  <nav area="linkchecker" navm="1">linkchecker/xml/lang_de_DE.xml;navigation/content/linkchecker/main</nav>
  <nav area="linkchecker">linkchecker/xml/lang_de_DE.xml;navigation/content/linkchecker/overview</nav>
  <nav area="lc_whitelist">linkchecker/xml/lang_de_DE.xml;navigation/content/linkchecker/whitelist</nav>
</nav_sub>

```

6. *nav_sub*: Es besteht die Möglichkeit, das Plugin in die Navigation zu schreiben, dazu erwartet der „nav_sub“-Tag die öffnenden und schließenden „nav“-Tags, die die Eigenschaft „area“ (Name der passenden Area) erwartet sowie als Inhalt den relativen Pfad zur Navigations-XML-Datei (Beispiel: „linkchecker/xml/lang_de_DE.xml;navigation/content/linkchecker/main“). Zusätzlich gibt es die optionale Eigenschaft „navm“, die den Sub-Navigationspunkt einer Navigation zuweist (Beispiel: 1 für den Navigationsbereich „Content“). Eine weitere Eigenschaft ist „menuless“; der Plugin Manager setzt automatisch den Wert 1.