

Plugin Manager

Version 1.0.1 in der Fassung CONTENIDO 4.9.0 RC1

Stand: 1. März 2013

Verfasser: Frederic Schneider <frederic.schneider@4fb.de>

Grundüberlegungen

Plugins sind ein wichtiger Bestandteil von CONTENIDO. Sie erweitern die Funktionalität des Grundsystemes. Neben vielen Plugins aus der Community gibt es einige, von Haus aus mit CONTENIDO 4.9 ausgelieferte Plugins wie den Linkchecker oder das AMR-Plugin von Murat Purc. Der Plugin Manager (im Folgenden „PIM“ abgekürzt) ermöglicht, diese Plugins standardisiert installier- und deinstallierbar zu machen. Hierzu wurde eine einheitliche Konfigurationsdatei erarbeitet, die alle notwendigen SQL-Einträge anlegt, um ein Plugin im Backend von CONTENIDO verwenden zu können.

Funktionsweise des PIM

Wir betrachten in dieser Dokumentation vor allem Plugins, die eine Auswirkung auf das Backend haben respektive dessen Dateien im Ordner „/contenido/plugins/“ abgelegt werden.

Damit ein Plugin über den PIM installiert werden kann, muss es in einem mit Zip komprimierten Archiv vorliegen: *[namedesplugins].zip*.

In diesem Zip-Archiv kann der Plugin-Autor alle relevanten Dateien (PHP-Dateien, Template-Dateien, Dateien zur Lokalisierung et cetera) seines Plugins ablegen. Die Dateien werden mit der Installation durch den PIM in den CONTENIDO-Plugin-Ordner abgelegt.

Des Weiteren erwartet der PIM mindestens eine Konfigurationsdatei, die den Namen *plugin.xml* trägt (auf die Eigenheiten wird im Laufe des Dokumentes detailliert eingegangen). Diese Datei ist unter anderem dafür zuständig, den Ordernamen des zu installierenden Plugins zu definieren, ebenso aber festzulegen, welche Einträge in den CONTENIDO-Datenbanktabellen „actions“, „area“, „files“, „frame_files“ sowie „nav_sub“ angelegt werden.

Neben der *plugin.xml* ist es optional möglich, eine *plugin-install.sql* sowie die dazugehörige *plugin-uninstall.sql* in das Installations-Archiv abzulegen. Diese SQL-Dateien ermöglichen einem Plugin-Entwickler, Plugin-spezifische Tabellen in der Datenbank anzulegen (Details im Laufe des Dokumentes).

Sobald ein Nutzer im Backend von CONTENIDO unter den Reitern „Administration“ -> „Plugin Manager“ ein mit Zip komprimiertes Archiv auswählt und auf „Plugin-Paket hochladen“ klickt, extrahiert der PIM die darin befindlichen Dateien in den temporären CONTENIDO-Ordner („/data/temp/“). Im nächsten Schritt überprüft der PIM, ob der Inhalt der *plugin.xml*-Datei dem Standard entspricht (Validierung). Ebenso wird überprüft, ob das Plugin für die installierte CONTENIDO-Version Gültigkeit besitzt. So ist es einem Plugin-

Entwickler möglich zu definieren, dass ein Plugin bspw. erst ab der Version 4.9.0 Beta 1 und maximal für die Version 4.9.1 installierbar ist.

Besonderheit „Noch nicht installierte Plugins“

Wie im Kapitel „Grundüberlegungen“ dargelegt, liefert CONTENIDO von Haus aus spezielle Plugins mit (Stand: Version 4.9.0 Beta 1):

- Advanced Mod Rewrite
- Content Allocation
- Cronjobs Overview
- Linkchecker
- Newsletter
- Smarty Wrapper
- URL Shortener
- Workflow

In den Versionen vor Version 4.9.0 Beta 1 waren diese Plugins fest installiert bzw. konnten teilweise in der Setup-Routine von CONTENIDO ausgewählt werden. Mit der Einführung des PIM sind alle Plugins standardmäßig nicht mehr installiert. Durchaus liegen alle Dateien der Plugins jedoch weiterhin im CONTENIDO-Plugin-Ordner vor. Sofern ein Plugin nicht in der Datenbanktabelle „plugins“ eingetragen wurde und in einem solchen Ordner eine *plugin.xml*-Datei vorliegt, werden diese Plugins im PIM unter dem Punkt „Noch nicht installierte Plugins“ angezeigt. Plugins werden in dieser Kategorie seit Version 4.9.0 RC 1 auch aufgeführt, nachdem sie im Backend hochgeladen wurden. Mit einem Klick auf das jeweilige Installations-Icon ist es dem Administrator möglich, diese Plugins zu installieren.

Durch die in der *plugin.xml*-Datei definierten Inhalte erzeugt der PIM sodann automatisiert Einträge in den CONTENIDO-Datenbanktabellen. Um eine spätere rückstandslose Deinstallation zu ermöglichen, merkt er sich, unter welcher Id in der Tabelle „area“ ein Plugin installiert wurde. Die Standard-Informationen eines Plugins werden in der mit Version 4.9.0 Beta 1 erweiterten Datenbanktabelle „plugins“ abgespeichert, hierzu gehören etwa der Name eines Plugins oder dessen Ordernamen im CONTENIDO-Plugin-Ordner.

Unter der Rubrik „Noch nicht installierte Plugins“ liegen zudem all jene Plugins vor, die aus der Datenbank wieder entfernt wurden (vgl. Punkt „Besonderheit Deinstallation“). Um ein Plugin auch aus dem Dateisystem zu entfernen, also vollständig zu deinstallieren, ist hierzu ein Klick auf das „Löschen“-Icon notwendig.

Besonderheit „Aktualisieren“ eines Plugins

Es ist möglich ein Plugin zu aktualisieren. In der bisherigen Version des PIM wird hierzu ein Plugin ausgewählt, auf „Aktualisieren“ geklickt und ein mit Zip komprimiertes Archiv ausgewählt. Der PIM überprüft, ob das zur Aktualisierung ausgewählte Plugin das gleiche ist wie das bereits installierte. Der PIM geht im Rahmen der Aktualisierung so vor, dass er das bisherige Plugin wie unter „Besonderheit Deinstallation“ formuliert rückstandslos entfernt und danach wieder neu installiert.

Wichtig: Durch die rückstandslose Deinstallation werden auch die individuell angelegten Plugin-Tabellen entfernt, selbstredend inklusive deren Inhalte.

Tipp für Plugin-Entwickler: Sofern es im Sinne eines Anwenders ist, dass die Einträge in der Plugin-spezifischen Tabelle nicht entfernt werden und mit dem Update eines Plugins sich lediglich PHP-Dateien geändert haben, wäre es durchaus denkbar, ein spezielles mit Zip komprimiertes Update-Archiv anzubieten, das ohne eine *plugin-install.sql* und eine *plugin-uninstall.sql* auskommt. In diesem Fall „tauscht“ der PIM lediglich die Dateien (PHP, Templates usw.) aus und legt die Einträge unter den Datenbanktabellen „actions“, „area“ et cetera neu an. Genauso könnte in einem speziellen Update-Archiv auch eine andere *plugin-install.sql* als im Installations-Archiv liegen, die etwa nur *ALTER TABLE*-Befehle ausführt, sofern sich lediglich an der Datenbanktabellenstruktur etwas geändert hat.

Besonderheit Deinstallation

Wird ein installiertes Plugin deinstalliert, so entfernt der PIM rückstandslos alle entsprechenden Einträge in der Datenbanktabelle (nicht die Plugin-Dateien). Hierzu speichert der PIM schon bei der Installation die Verknüpfung eines Plugins mit der Datenbanktabelle „area“. Diese Verknüpfung wird in der Datenbanktabelle „plugins_rel“ dokumentiert. Durch die Definition in der *plugin-uninstall.sql* können auch vom Plugin individuell angelegte Tabellen entfernt werden. Wichtig: Etwaige spezielle Anpassungen oder Konfigurationen gehen mit der Deinstallation des Plugins verloren!

Aufbau der plugin.xml

Die plugin.xml ist in zwei Abschnitte eingeteilt:

- Die allgemeinen Plugin-Informationen, zu finden innerhalb des XML-Tags „general“
- Die Datenbank-spezifischen Definitionen für die Datenbanktabellen „actions“, „area“, „files“, „frame_files“, „nav_sub“, zu finden innerhalb des XML-Tags „contenido“

Vorbemerkung

Die XML-Dateistruktur ist in der XSD-Datei *plugin_info.xsd* definiert, zu finden im CONTENIDO-Ordner „/contenido/plugins/pim/xml/“. Diese XSD-Datei ist auch die Grundlage für die Validierung eines Plugins im Rahmen der Installations-Routine.

Allgemeine Plugin-Informationen

In diesem Abschnitt innerhalb des „general“-Tags von XML werden allgemeine Plugin-Informationen definiert, die später in der Datenbanktabelle „plugins“ abgelegt werden. Sie sind wichtig, um ein Plugin zuordnen und installieren zu können. Aber auch, um in der Backend-Ansicht des PIM in CONTENIDO Informationen über das installierte Plugin anzeigen zu können.

Übersicht über die im „general“-Tag vorkommenden Felder:

- active-Tag: Hat den Wert 1 (aktiviert) oder 0 (nicht aktiviert). Nicht aktivierte Plugins werden im Backend nicht geladen, können also nicht verwendet werden. Der Standardwert ist 1 (aktiviert).
- plugin_name: Hierbei handelt es sich um die Bezeichnung eines Plugins. Pflichtfeld.
- plugin_foldername: Hierbei handelt es sich um den Ordnername des Plugins. Der so benannte Ordner wird unter „/contenido/plugins/“ angelegt. Der PIM prüft bei der Installation eines Plugins, ob es diesen Ordner schon gibt. Er muss deshalb einmalig gewählt werden. Pflichtfeld.
- uuid: Eine UUID ist eine von der four for business AG zu vergebende einmalige Identifikationsnummer eines Plugins. Dies ist im Besonderen dafür wichtig, um bei

einer Installation bzw. einer Aktualisierung zu überprüfen, ob das Plugin bereits installiert wurde. Eine UUID kann unter der Internetadresse [LINK FOLGT] generiert werden. Hierzu werden die in der *plugin.xml* definierten Felder „copyright“ sowie „plugin_foldername“ benötigt. Pflichtfeld.

- description: Beschreibung eines Plugins, relevant lediglich für die Backend-Ansicht des PIM in CONTENIDO. Optionales Feld.
- author: Name des oder der Autore(s)(n) eines Plugins, relevant lediglich für die Backend-Ansicht des PIM in CONTENIDO. Pflichtfeld.
- copyright: Copyright-Hinweis eines Plugins, ggfls. identisch mit dem Feld „author“. Pflichtfeld.
- mail: E-Mail-Adresse der Plugin-Autoren, relevant lediglich für die Backend-Ansicht des PIM in CONTENIDO. Optionales Feld.
- website: Website zu den Plugin-Autoren, relevant lediglich für die Backend-Ansicht des PIM in CONTENIDO. Optionales Feld.
- version: Versionsbezeichnung des Plugins. Hierzu ist eine Versionierungsbezeichnung zu wählen, die mit der PHP-Funktion *version_compare()* kompatibel ist. Pflichtfeld.
- min_contenido_version: In diesem Feld wird definiert, welche Version von CONTENIDO ein Administrator mindestens installiert haben muss, damit das Plugin lauffähig ist. Pflichtfeld.
- max_contenido_version: In diesem Feld wird definiert, welche Version von CONTENIDO ein Administrator maximal installiert haben darf, damit das Plugin lauffähig ist. Ggfls. ist ein Plugin mit Version 4.9.0 Beta 1 lauffähig, aber nicht mit Version 4.9.0 RC 1. In diesem Fall würde die Definition „4.9.0-beta1“ richtig sein. Optionales Feld.

Datenbank-spezifische Definitionen im „contenido“-Tag

Sofern ein Plugin nicht im Backend aufrufbar ist, kann dieser Komplex weggelassen werden. Die Bezeichnungen aller Attribute in diesem Abschnitt entsprechen den jeweiligen Spaltennamen der Datenbanktabellen.

Übersicht über die im „contenido“-Tag vorkommenden Felder:

- areas: Hierbei werden Einträge in der Datenbanktabelle „area“ angelegt. Der PIM erwartet die optionale Angaben der Attribute „parent“ und „menuless“. „parent“ entspricht entweder dem Namen einer anderen Area oder besitzt den Wert „0“ (hat also keinen Vaterwert). „menuless“ besitzt entweder den Wert „0“ (Standard) oder „1“ – sprich: entweder eine Area mit einem linken Menü-Frame oder ohne.
Beispiel: `<area parent="linkchecker" menuless="1">lc_whitelist</area>`
- actions: Hierbei werden Einträge in der Datenbanktabelle „actions“ angelegt. Der PIM erwartet die Pflicht-Angabe des Attributes „area“. Dieser Wert entspricht dem Namen einer Area in der Datenbanktabelle „area“.
Beispiel: `<action area="linkchecker">linkchecker</action>`
- frames: Der Frames-Tag umfasst die beiden Datenbanktabellen „files“ und „frame_files“. Der PIM erwartet die Pflicht-Angabe des Attributes „area“, dieser Wert entspricht dem Namen einer Area in der Datenbanktabelle „area“. Als zweites Attribut steht optional „filetype“ zur Verfügung, der Standardwert lautet „main“ (der zweite, mögliche Wert ist „inc“). Als drittes Attribut ist verpflichtend „name“ vorgesehen, dies entspricht einem relativen Pfad zu einer Frame-Datei. Als viertes

Attribut ist optional die „frameld“ anzugeben, dies entspricht den vier Backend-Frames von CONTENIDO: „1“, „2“, „3“ oder „4“.

Beispiel: `<frame area="linkchecker" filetype="main"`

`name="linkchecker/includes/include.linkchecker.php" frameld="4" />`

- **nav_sub:** Hierbei werden Einträge in der Datenbanktabelle „nav_sub“ angelegt. Der PIM erwartet die Angabe der Pflicht-Attribute „area“ und „level“. „area“ entspricht dem Namen einer Area in der Datenbanktabelle „area“. „level“ entspricht dem Level-Wert „0“ oder „1“. Als drittes, optionales Attribut kann „navm“ angegeben werden, dies entspricht den Navigation-Menü-Ids im Backend von CONTENIDO.

Beispiel: `<nav area="linkchecker" level="0"`

`navm="1">linkchecker/xml/lang_de_DE.xml;navigation/content/linkchecker/main</nav>`

- **nav_main:** Über den Tag „nav_main“ ist es möglich, Einträge in der Datenbanktabelle „nav_main“ vorzunehmen (Menüpunkte im obersten Level)

Syntax: `<nav_main><nav>locationname</nav></nav_main>`

plugin_dependencies

Interessierte Plugin-Entwickler werden in der *plugin_info.xsd*-Datei das Vorkommen eines „plugin_dependencies“-Tags entdecken. Hierbei ist vorgesehen, dass ein Plugin nur dann installiert werden kann, wenn ein anderes Plugin im Vorkommen einer bestimmten Version bereits installiert ist. Diese Funktion ist jedoch bislang nicht implementiert.

Plugin-spezifische Datenbanktabellen

Den Plugin-Entwicklern ist es möglich, eigene Datenbanktabellen anzulegen. Hierzu sind Einträge in der *plugin-install.sql* notwendig. Diese Datei ist im Hauptordner des mit Zip komprimierten Archives abzulegen.

Die *plugin-install.sql* erwartet pro Zeile ein SQL-Statement. Auf zwei oder mehrere Zeilen verteilte Statements sind ungültig und führen zu einem Fehler. Ein SQL-Statement in dieser Datei darf ausschließlich mit folgenden SQL-Befehlen beginnen:

- CREATE TABLE IF NOT EXISTS
- INSERT INTO
- UPDATE
- ALTER TABLE

Für den Tabellennamen ist das Präfix „!PREFIX!“ zu verwenden. Hierbei setzt der PIM automatisch das vom Administrator definierte SQL-Präfix (i. d. R. „con“) und den Wert „_pi“ ein. Aus „!PREFIX!“ wird also beispielsweise „con_pi“. Ein SQL-Statement könnte dann zum Beispiel folgendermaßen aussehen: *INSERT INTO !PREFIX!_news_log (...)*. SQL-Statements ohne das Präfix werden nicht ausgeführt.

Analog zu der *plugin-install.sql* gibt es für die Deinstallations-Routine die *plugin-uninstall.sql*. Diese erlaubt die beiden SQL-Befehle:

- DELETE FROM
- DROP TABLE

Es ist entsprechend das Präfix „!PREFIX!“ zu verwenden.

Änderungen von Version 4.9.0 Beta 1 zu Version 4.9.0 RC 1

- Implementierung einer Aktivierungs- / Deaktivierungsfunktion für einzelne Plugins („active“-Wert in der Datenbanktabelle)
- Implementierung des „nav_main“-Tags für Navigationspunkte
- Wird ein installiertes Plugin entfernt, so werden lediglich die Datenbankinträge gelöscht. Um das Plugin vollständig auch aus dem Dateisystem zu entfernen, muss es unter dem Punkt „Noch nicht installierte Plugins“ gelöscht werden.
- Ein Plugin wird beim Hochladen nicht mehr direkt installiert, sondern lediglich auf dem Dateisystem gespeichert (die diversen, in der *plugin.xml* definierten Tests finden jedoch auch hier bereits statt). Um ein Plugin vollständig zu installieren, also auch Einträge in der Datenbank vorzunehmen, muss es unter dem Punkt „Noch nicht installierte Plugins“ aktiviert werden.
- Die Xml-Felder „actions“, „frames“, „nav_main“ sowie „nav_sub“ sind optional, werden also nicht für die Installation eines Plugins vorausgesetzt (lediglich mind. eine „area“ wird erwartet)
- Das Xml-Konfigurationsfeld „copyright“ ist ein Pflichtfeld
- Fehlermeldung bei ungültigen Xml-Dokumenten (*plugin.xml*) wird nicht mehr als Exception angezeigt, sondern als Fehlermeldung über das PIM-Fehlermeldungstemplate
- Ergänzung von Tooltips unter „Noch nicht installierte Plugins“
- Plugins werden in der Übersicht alphabetisch sortiert
- Bei der Installation eines Plugins werden jetzt auch Plugin-spezifische Einträge der Tabelle „nav_sub“ in der Relations-Tabelle abgespeichert