

NAME

fetchnews – holen und senden von News via NNTP

ÜBERSICHT

fetchnews [-q] [-v] ... [-x *Zahl*] [-l] [-n] [-f] [-P] [-w]

BESCHREIBUNG

Leafnode ist ein USENET-Software-Paket, das für kleine Sites gedacht ist, wo es wenige User und nur geringe Festplatten-Kapazität gibt, aber trotzdem eine große Zahl von Gruppen gewünscht wird.

Leafnode ist so geschrieben, daß es Fehler selbständig reparieren und keinerlei manuelle Pflege benötigen soll.

Fetchnews (das bis Version 1.9.2 "fetch" hieß) ist das Programm, das neue Artikel vom Upstream-Server lädt und lokal geschriebene Artikel an diesen weiterreicht.

OPTIONEN

- q unterdrückt einige Warnungen und hebt zuvor gesehene -v-Schalter auf.
- v hebt die Wirkung voriger -q-Schalter auf und veranlaßt **fetchnews**, mehr Ausgaben zu produzieren (verbose = wortreich). Maximal vier -v können angegeben werden, je mehr -v, desto mehr Text wird ausgegeben:
- v Zeigt unkritische Fehler an.
- vv Zeigt alle Gruppen-Befehle an
- vvv Zeigt alle Artikel-Befehle an.
- vvvv Für Debugging-Zwecke. Erwarten Sie nicht, daß die zusätzlichen Ausgaben Sinn ergeben.
- w Warten. Damit laufen die XOVER-Aktualisierungen (nach der Abholphase) im Vordergrund ab. Ohne diese Option laufen die XOVER-Aktualisierungen auf Systemen, die fork(2) unterstützen, im Hintergrund. Nützlich zum Debugging, oder um bei manuellem Start Unordnung in der Terminalanzeige zu vermeiden.
- x *Zahl* Gehe *Zahl* Artikel zurück und lade fehlende Artikel vom Upstream-Server. Das hilft, wenn der Upstream seine neuen Artikel nicht in Reihenfolge der Artikelnummer erzeugt oder wenn seine Übersicht (NOV/XOVER)-Daten nicht mit den Gruppeninhalten synchronisiert sind.
- l Verwende nur den ersten Upstream-Server (l = local). Dies erhöht die Geschwindigkeit des Vorganges erheblich.
- n Keine Newsgruppen abbestellen. Technisch: Lösche die Verknüpfungen im Verzeichnis */opt/leafnode-1/var/spool/news/interesting.groups* nicht.
- f Löschen der Datei *active.read*. Dadurch wird die Liste der verfügbaren Newsgruppen vom Upstream-Server neu geladen (force). Dies sollte nur selten geschehen, da es einige Zeit benötigt. Siehe auf *timeout_active* in **leafnode(8)**.
- P Sende nur die Artikel, die sich in */opt/leafnode-1/var/spool/news/out.going* befinden und lade keine neuen Artikel vom Server.

RÜCKGABEWERT

fetchnews gibt den Wert 0 zurück, wenn es alle Server erreichen konnte und keine weiteren Fehler aufgetreten sind. Es gibt 2 zurück, wenn mindestens einer der Server nicht erreichbar waren, und 1 bei sonstigen Fehlern.

DATEIEN UND KONFIGURATION

/opt/leafnode-1/var/spool/news/leaf.node/active.read

Diese Datei wird nach erfolgreichem Laden der Liste der verfügbaren Newsgruppen von den Upstream-Servern angelegt. Fehlt sie, so werden sie analog zum -f Parameter geladen.

Bitte lesen sie in **leafnode (8)** nach, um eine Aufstellung der weiteren betroffenen Dateien und der Konfigurationsmöglichkeiten von fetchnews zu erhalten.

UMGEBUNGSVARIABLEN ("ENVIRONMENT")

LN_SKIP_GROUPS=*pattern[,pattern[...]]* (seit v1.9.53)

Alle Gruppen, die einem der **wildmat(3)**- Muster *pattern* entsprechen, werden übersprungen. Damit können zum Beispiel Binärgruppen tagsüber schnell ausgelassen werden. Beispiel:

```
env 'LN_SKIP_GROUPS=*binary*,alt.*' /opt/leafnode-1/sbin/fetchnews -nq
```

LN_LOCK_TIMEOUT

Diese Variable wird als nicht-negative ganze Zahl gelesen und gibt an, wie viele Sekunden fetchnews auf die Lock-Datei warten soll, 0 bedeutet, unbegrenzt zu warten. Diese Variable genießt Vorrang vor der Konfigurationsdatei.

ALGORITHMUS

Dies ist eine kurze Beschreibung dessen, was **fetchnews** tut.

Zuerst sendet **fetchnews**, nachdem eine Verbindung mit dem Upstream-Server hergestellt wurde, alle Artikel, die sich in */opt/leafnode-1/var/spool/news/out.going* befinden und Leserechte für den Benutzer besitzen (u+r, owner/user read permission), mittels **POST**. Schlägt dieses Kommando fehl, wird der betreffende Artikel in das Verzeichnis */opt/leafnode-1/var/spool/news/failed.postings* verschoben, das der Systemadministrator von Zeit zu Zeit aufräumen sollte. Die Dateien zurück nach */opt/leafnode-1/var/spool/news/out.going* zu verschieben reicht aus, um die entsprechenden Artikel beim nächsten Lauf von **fetchnews** zu versenden. Das Ausführungsrecht (u+r) wird abgefragt, um das Versenden unvollständiger Artikel zu verhindern, die **leafnode(8)** zur gleichen Zeit empfängt. Wenn **POST** erfolgreich ist, wird der Artikel aus *out.going* entfernt, um ihn nicht zweifach in eine moderierte Newsgruppe eines anderen Servers zu posten (was dem Moderator mehrere Kopien des Artikels zustellen würde). **Hinweis:** ältere fetchnews-Versionen haben versucht, auf allen Servern zu posten, dies ist aber dann fehlgeschlagen, wenn nicht alle Server erreicht werden konnten. Wenn Sie einen etwas fragwürdigen Upstream-Server verwenden wollen, der beispielsweise schnell ist, aber Ihre Postings nicht zuverlässig weiterleitet, können Sie fetchnews verbieten, auf diesem Server zu posten, indem sie in der Konfigurationsdatei die Server-Option "nopost" verwenden, siehe **leafnode(8)**.

Als nächstes empfängt **fetchnews** die Liste der vorrätigen Gruppen des Upstream-Servers mittels "LIST ACTIVE" und "LIST NEWSGROUPS" und fertigt daraus die von **leafnode** an (falls der Zeitraum timeout_active seit dem letzten Laden der Liste vergangen ist oder die Datei active.read nicht vorhanden ist; ansonsten wird die Liste mittels "NEWSGROUPS" aktualisiert).

Zuletzt durchläuft **fetchnews** die Liste der Newsgruppen, indem es ein "GROUP-", ein XOVER (falls vom Server unterstützt) oder "XHDR Message-ID"- und eine Anzahl von "HEAD"- und "BODY"- oder "ARTICLE"-Kommandos für jede Gruppe ausführt, die in letzter Zeit gelesen wurde. Dabei werden nur Artikel geholt die nicht älter als "maxage" Tage alt sind. Dabei wird maxage kleiner als oder gleich expire bzw. groupexpire angenommen.

ANMERKUNGEN

Fetchnews benutzt kein "NEWNEWS". "NEWNEWS" wird nicht eingesetzt, da dies oft eine große Belastung für den Upstream-Server darstellt.

Die einzige Möglichkeit, zu kontrollieren, welche Newsgruppen lokale Benutzer lesen können, besteht darin, welche Gruppen **fetchnews** beim Upstream-Server lesen kann. Dies beschränkt allerdings keine

Crosspostings. Wenn Ihre Site groß genug ist, daß Sie den Zugriff auf Newsgroups kontrollieren müssen, sollten Sie sich überlegen, einen Newsserver einzusetzen, der für große Sites geplant ist.

Fetchnews setzt seine UID auf "news", falls es von root gestartet wurde, und bricht ab, wenn es UID "news" nicht bekommen kann.

FEHLER

Fetchnews ignoriert "Supersedes:"- und "Control:"-Header.

Fetchnews löscht niemals eine Gruppe aus der Liste der aktiven Newsgroups.

ENTWICKLUNG

Ältere Versionen von **fetchnews** haben alle Artikel in out.going verschickt. Seit Version 1.9.23 versendet **fetchnews** nur noch Artikel in Dateien, deren Ausführungs-Bit (executable) gesetzt ist. **Leafnode** setzt dieses Bit nach einem vollständigen POST. Diese Änderung verhindert, daß unvollständige Artikel verschickt werden, falls **leafnode** einen Artikel erhält, während **fetchnews** gerade sendet.

AUTOR

Die ersten Versionen von fetchnews wurden von Arnt Gulbrandsen <agulbra@troll.no> unter dem Namen fetch geschrieben; Copyright 1995 Troll Tech AS, Postboks 6133 Etterstad, 0602 Oslo, Norway, fax +47 22646949.

Weiterentwickelt wurde fetch von Cornelius Krasel <krasel@wpxx02.toxi.uni-wuerzburg.de>, Randolph Skerka <Randolf.Skerka@gmx.de> und Markus Enzenberger <enz@cip.physik.uni-muenchen.de>. Copyright (an den Veränderungen) 1997-1999.

Modifiziert von Matthias Andree <matthias.andree@gmx.de>. Copyright 2002 - 2004.

Deutsche Übersetzung: Alexander Reinwarth <a.reinwarth@gmx.de>, ergänzt von Ralf Wildenhues.

SIEHE AUCH

leafnode(8), **newsq(1)**.