

RPM tranzakciók és visszavonás

Ha egy frissítés jól működik, néhány új képességhez, vagy jobb teljesítményhez jutunk. Ha azonban becsúszik egy hiba, az egész hétvégénk tönkremehet. Nézzük meg, hogyan térhetünk vissza a korábbi verzióhoz és tarthatjuk karban rendszerünket.

Hányszor fordult már elő, hogy valami csodásnak ígérkező új programot telepítettünk fel, csak azért, hogy szinte azonnal kiderüljön: valójában egyáltalán nem is akartuk felrakni? Sőt, a dolog még kellemetlenebb lehet, ha a feltelepítendő programhoz jó néhány egyéb csomagot kell frissítenünk további párat pedig az alaptól felraknunk. Ha a dolgokat az eredeti állapotba szeretnénk visszaállítani, több forrásból kellene előkeresnünk a továbbfejlesztett csomagok korábbi verzióját, visszalépni ezekre a verziókra, valamint leszedni minden újonnan felrakott csomagot. Természetesen ha nem tartjuk rendszeresen nyilván milyen új csomagokat változtattunk meg és mi is volt a korábbi verzió, a dolgok még rosszabbra fordulhatnak. Hát nem lenne sokkal egyszerűbb, ha csak megnyomnánk egy gombot, vagy kiadnánk egy parancsot, és máris visszatérhetnénk a régi verzióhoz?

Néhány környezetben a frissítés előtti állapothoz való visszatérés nem csupán lehetőség, hanem egyenesen követelmény. A telekommunikációs cégek felszerelésének fejlesztésekor például a programozó és alkatrész szolgáltató cégeknek minden fejlesztést egy karbantartási ablaknak nevezett, korlátozott időszakban kell elvégezniük. Ugyan ebben a karbantartási ablakban a továbbfejlesztés során bekövetkezett valamennyi változtatást vissza is kell tudniuk állítani. Ha nem sikerül a karbantartási ablakon belül befejezni a műveletet, annak súlyos pénzügyi következményei lehetnek.

Frissítés és helyreállítás, nehézkesen

Bármennyire is kívánatos lenne az RPM-ek automatikus visszaállítása egészen mostanáig nem volt erre lehetőségünk. Az igazság kedvéért hozzá kell tennünk, hogy az RPM támogatja a csomagkészletek visszafejlesztését (*downgrade*). Például, ha a *foo-1-1* RPM csomagról a *foo-1-2* verzióra fejlesztettük, az `rpm` parancsot az `--oldpackage` kapcsolóval futtatva visszaléphetnénk a korábbi verzióhoz; valahogy így:

```
# rpm -uvh --oldpackage foo-1-1.i386.rpm
Preparing... #####
[100%]
```

```
Upgrading...
1:foo ##### [100%]
```

Amennyiben a *foo-1-2* fejlesztéséhez nem volt szükségünk semmilyen további RPM csomagra, az `--oldpackage` kapcsoló kiválóan működik. Mindössze a korábbi *foo-1-1* RPM csomagot kell megtalálnunk és kész is vagyunk. Azonnal megváltozik a helyzet azonban, ha a *foo-1-2* más csomagoktól is függ, amelyeket így szintén fel kellett tennünk. Ez esetben elő kell keresnünk ezeket az RPM-eket különféle helyekről – a telepítő lemezről, a terjesztésünk hibajegyzék oldaláról, egyéb RPM tárhelyről vagy a projektek web-lapjairól.

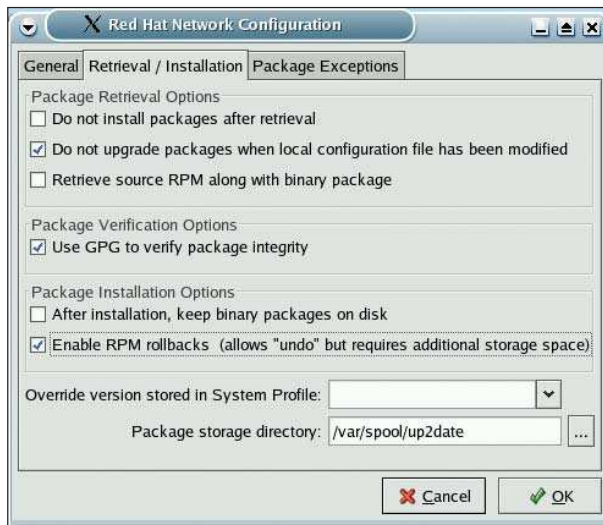
Miután levadásztuk az összes szükséges RPM-et, valamennyit vissza kell fejlesztenünk majd letörölnünk a frissen telepítetteket. Ha ezt fordított sorrendben csinálnánk, letörölnénk a frissen feltelepített RPM-eket és csak aztán fejlesztenénk vissza a frissített RPM-eket, akkor az RPM hibáüzenetekkel örvendeztetne meg bennünket, miszerint ezekre a csomagokra a *foo-1-2* csomagnak szüksége van. Tömören: ez idáig több RPM csomag visszaállítása igencsak nehézkes volt, és tele hibalehetőségekkel.

Tranzakció alapú visszavonás és rendszermentés

2002 elején, *Jeff Johnson*, az RPM aktuális gazdája, megpróbált ellenszert találni erre a visszavonási problémára és az RPM 4.0.3-as változatában bevezette a tranzakció alapú visszavonási képességet. Ez már magában hordozta az RPM csoportok automatikus visszavonásának lehetőségét. Akárcsak más új képességek, ez is kicsit elnagyoltra sikerült, és – az RPM levelezési (rpm-list@redhat.com) lista néhány levelétől eltekintve -, teljesen dokumentálatlan volt. Az elmúlt másfél évben a tranzakció alapú visszavonás folyamatosan fejlődött. A Red Hat 9-el érkező, jelenlegi 4.2-es RPM kiadásban például már teljesen használható ez a funkció.

A tranzakció alapú RPM visszavonás működési elve

A színpalak mögött az RPM minden feltelepített RPM csoportot egy-egy tranzakciónak tekint. Ez igaz, ha egyetlen RPM-et telepítünk fel (egy RPM tranzakciója) és akkor is, ha több RPM-et teszünk fel egyszerre. Minden ilyen tranz-



1. ábra RPM visszavonás engedélyezése az up2date programban

akció egyedi tranzakció azonosítót (TID) kap. Ahogy az RPM felkerül és frissül, bejegyzését az RPM adatbázisban megjelöljük a telepítéskor felhasznált TID azonosítóval. Ezáltal az RPM nyomon tudja követni, hogy mely RPM csomagokat mely tranzakciók során telepítettük és frissítettük. Ha az RPM visszavonja a tranzakciócsoportot, el kell tudnia érni a tranzakció végrehajtása idején a rendszeren lévő csomagokat. Ezt a problémát úgy oldja meg, hogy törlés előtt valamennyi csomagot újracsomagolja, majd eltárolja őket a *repackage* könyvtárban (alapértelmezés szerint, */var/spool/repackage*). Az újracsomagolt csomagok az RPM által kezelt és a törlés pillanatában a rendszeren található valamennyi állományt, a régi RPM-ek fejléceit illetve a régi RPM-el kapott parancsfájlocskákat (*scriptlets*) tartalmazzák. Felmerül a kérdés, ugyan mi köze mindennek a frissítésekhez? Ha ugyanis frissítünk egy RPM csomagot, attól még nem töröljük le – nem igaz? De bizony, letöröljük. Ugyanis az RPM frissítése két részből áll: az új csomagot feltelepítjük, a régit pedig letöröljük. Ennek megfelelően, valahányszor frissítünk egy csomagcsoportot, az RPM előbb újracsomagolja az összes frissítendő csomagot, ezután telepíti az újakat, végül pedig letörli a régieket. Amikor az RPM újracsomagolja a régi csomagokat, egyúttal meg is jelöli azokat a futó tranzakció TID azonosítójával. Ennek köszönhetően többé nem kell a netet, tárolókat, vagy mentéseket böngészni a frissített csomagokat keresendő. Mivel az újracsomagolt csomagok azokat az állományokat tárolják amelyek a frissítés idején a rendszerünkön voltak, a beállításfájlokat sem kell mentésekből visszaállítanunk. Mellékhatásként az újracsomagolt csomagban található állományok md5 ellenőrzőösszegei valószínűleg hibásak lesznek, ugyanis az RPM ezeket nem számítja ki újra az újracsomagolt csomag létrehozása során. Szerencsére ez a tranzakciók visszaállításakor nem okoz gondot az RPM-nek, de ha közvetlenül szeretnénk használni csomagokat használnunk kell a *--nodigest* kapcsolót.

Az RPM-nek a *repackage* könyvtár feltöltése után a csomagok visszaállításához már csak a visszaállítás célpontjára (azaz a dátumra, amelyre vissza kell állni) van szüksége.

Az RPM a TID alapján el tudja dönteni milyen tranzakciók történtek rendszerünkön a visszaállítási időpont óta. Ez után az RPM veszi ezeket a tranzakciókat, sorba rendezi a legfrissebbtől a legkevésbé frissig, majd valamennyin végrehajtja a következő lépéseket:

- Kikeresi az összes ilyen TID jelet viselő újracsomagolt csomagot.
- Kikeresi az összes jelenleg feltelepített csomagot amelyet ilyen TID jelöl de nincsen megfelelő újracsomagolt csomagja.
- Felépíti a visszaállítási tranzakciót. Az újracsomagolt csomagok ebben mint telepítendő szerepelnek, a megfelelő újracsomagolt csomag nélküli csomagok pedig törlendő elemként.
- Lefuttatja a frissen elkészített visszaállító tranzakciót.

Ezeket a műveleteket ismételve a legfrissebbtől a céldátumhoz legközelebbi vagy azzal egyező dátumú csomagig, az RPM végiglépked a visszaállítási időpont óta bekövetkezett valamennyi tranzakción és eltünteteti hatásukat.

Hogyan használhatjuk a tranzakció alapú RPM visszaállítást

Azt gondolhatnánk, hogy ez elég bonyolult folyamat, de a tranzakció alapú visszaállítás valójában rendkívül egyszerű. Példaként telepítsünk egyetlen RPM csomagot majd vonjuk vissza. A leglényegesebb dolog amire emlékeznünk kell, hogy valahányszor frissítünk vagy törölünk egy csomagot, soha ne felejtjük el előtte az RPM-nek kiadni a régi csomag újracsomagolását végző parancsot. Ehhez a *--repackage* kapcsolót kell használnunk:

```
# rpm -Uvh --repackage foo-1-2.noarch.rpm
Preparing... ##### [100%]
Repackaging...
 1:foo ##### [100%]
Upgrading...
 1:foo ##### [100%]
```

A kapcsoló hatására az RPM előbb újracsomagolja a régi csomagot, majd frissít az új változatra. Törléskor szintén meg kell adnunk a *--repackage* kapcsolót, valahogy így:

```
# rpm -e --repackage foo
```

Az RPM törléskor nem mutat semmilyen kimenetet, de ha törlés után belenézünk az újracsomagolási könyvtárba, megtalálhatjuk az újracsomagolt csomagokat.

Az RPM tranzakció visszavonására a *--rollback* kapcsolót használjuk, amelyet a visszavonás időpontja követ.

A visszavonás időpontja lehet valódi dátum vagy olyan kifejezés mint „egy órával ezelőtt” (a dátumértelmező ugyanolyan formátumot használ, mint a *cvst*(1) parancs *-D* kapcsolója). Így, ha már eltelt egy óra a *foo* telepítése óta, és úgy döntünk, még sincs rá szükségünk, a következő sort gépeljük be:

```
# rpm -Uvh --rollback `2 hours ago`
Rollback packages (+1/-1) to
```

```
Thu Jul 31 23:26:52 2003 (0x3f29ddfc):
Preparing... ##### 100%]
1:foo ##### [ 33%]
```

A Rollback packages (+1/-1) kimenet mutatja, hogy az RPM egy darab csomagot (az előző verziót) fog felvenni, és egy csomagot (a jelenleg telepített verziót) fog törölni.

Hogyan készítsünk visszavonás alapú rendszert?

A tranzakció alapú visszavonás csak annyira használható amennyire az újracsomagolt csomagjaink tárháza. A leggyorsabban úgy tudjuk elrontani a dolgot, ha valamit a --repackage kapcsoló nélkül frissítünk vagy törölünk. Márpedig saját tapasztalataim szerint rendkívül könnyű elfeledkezni erről a kapcsolóról. Ezért aztán ha tranzakció alapú visszavonást szeretnénk használni, érdemes úgy beállítani az RPM-et, hogy automatikusan újracsomagolja minden törlést. Állítsuk be a %_repackage_all_erasures makrót a 1-re a /etc/rpm/macros állományban. Ha a fájl nem létezik nyugodtan hozzuk létre:

```
%_repackage_all_erasures 1
```

Alapértelmezés szerint az RPM nem von vissza frissen telepített csomagokat; azaz nem töröl le olyan csomagokat, amelyek a frissítésünk időpontjában még nem voltak a rendszeren. Valószínűleg nem szeretnénk, ha ez lenne az alapértelmezett működési forma, úgyhogy meg kell mondanunk az RPM-nek, hogy engedélyezze a törlést. Állítsuk a %_unsafe_rollback makrót olyan dátumra, amelyen túl már nem szeretnénk, ha az RPM visszavonáskor teljes törlést végezne. Erre a célra például nagyon jól megfelel az az időpont amikor a teljes rendszert feltelepítettük. A dátumot *epoch* után másodpercben kell megadni. Az *epoch* óta eltelt másodpercek számát a date parancs segítségével állíthatjuk elő:

```
date --date="2003/8/1" +%s
1059710400
```

(a dátumot átírtuk magyar sorrendre, helyes Local beállítású UNIXon ennek mennie kell. – a szerk.)

Ha például azt szeretnénk megadni az RPM-nek, hogy a 2003/8/1 előtti csomagokat ne távolítsa el teljesen (lásd a fenti példában használt dátumot), a következő sort kell a /etc/rpm/macros fájlba írunk:

```
%_unsafe_rollback 1059710400
```

Egyetlen dolog maradt még, amit esetleg érdemes átállítani, mégpedig az, hogy hol tárolja az RPM az újracsomagolt csomagokat. Tehetjük ezt például azért, hogy elegendő

hellyel rendelkező partícióra helyezzük őket. Az újracsomagolási könyvtár megváltoztatásához a %_repackage_dir makrót kell az új csomaggyűjtő hely elérési útjára állítanunk:

```
%_repackage_dir /my_rp_repository
```

Készítettünk tehát egy rendszert, amely automatikusan újracsomagolja a törléseket (így se mi, se más nem felejt el), visszavonáskor törli az újonnan felrakott csomagokat (de nem törli az egész rendszerünket) és a nekünk megfelelő helyen tárolja az újracsomagolt állományokat.

RPM tranzakciók visszavonása up2date segítségével

Red Hat 9 alatt az RPM tranzakció alapú visszavonási módszerét használva az up2date program is támogatja a visszavonásokat. Amennyiben be akarjuk állítani a tranzakció alapú visszavonást mindössze az up2date-config programot kell lefuttatnunk, rákattintanunk a Retrieval/Installation fülre, majd kiválasztanunk az Enable RPM rollbacks jelölőnégyzetet (1. ábra). Magát az RPM-et az előző szakaszban leírtak szerint kell beállítanunk. Az RPM és az up2date program beállítása után rendszerünket az up2date segítségével frissítve, az RPM frissítés előtt elkészíti az RPM-ekhez tartozó újracsomagolt csomagokat. Az ismert visszavonási időpontok listáját a következő parancs segítségével kapjuk meg:

```
up2date --list-rollbacks
```

Hatására valami ilyen listát kapunk:

```
# up2date --list-rollbacks
install time: Sun Jul 27 20:49:55 2003
tid:1059353395
[-] goo-1.0-1.0:
```

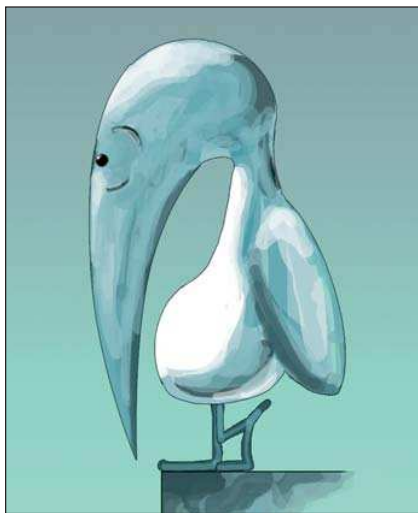
```
install time: Tue Jul 29 20:44:25 2003
tid:1059525865
[-] foo-1-2:
```

Ez a parancs akkor is jól jöhet, ha egyébként nem használjuk az up2date parancsot, ugyanis az rpm parancs nem képes ilyen adatok megjelenítésére.

A tranzakció visszavonását a --undo kapcsolóval végezzük, amely az utoljára telepített tranzakciót vonja vissza. Egszerűen adjuk ki a következő parancsot:

```
# up2date --undo
```

Amennyiben több tranzakciót szeretnénk visszavonni, futtassuk többször a parancsot. A grafikus felületen keresztül nem tudunk visszavonást kérni.



Automatikus visszavonás folt

Az RPM általában a legkisebb fáradság elvén dolgozik, azaz, ha egy vagy több RPM csomagot nem tud feltelepíteni, a tranzakcióban található egyéb csomagok ettől még felkerülnek. Bizonyos környezetekben ez a kívánatos viselkedés, máskor azonban sokkal jobb lenne ha automatikusan visszavonná a hibás tranzakciót. Minthogy én ilyen környezetben dolgozom (telekommunikáció), írtam egy automatikus visszavonás (*auto-rollback*) foltot. Ezzel a folttal az RPM-et úgy is be lehet állítani, hogy hibás tranzakció esetén automatikusan vonja vissza a hibás tranzakciót. Amennyiben az RPM a *%post* parancsfájlból okoz hibát, sajnos az RPM-et hátrahagyja; remélhetőleg ez hamarosan ki lesz javítva (van kedve valakinek foltozni?). Ha használni szeretnénk ezt a funkciót, a foltot (vagy a foltozott RPM-eket) a <http://www.lee.k12.nc.us/~joden/misc/patches/rpm> címről tölthetjük le. Az automatikus visszavonás folttal rendelkező RPM rendszerünk telepítése után be kell állítanunk az RPM-et, hogy használja az automatikus visszavonás képességet. Ehhez az */etc/rpm/macros* állományt kell átszerkeszteni, felvéve a következő makródefiníciót:

```
%_rollback_transaction_on_failure 1
```

Ezután ha majd legközelebb telepítünk illetve frissítünk egy RPM csoportot, és az egyik hibásan sikerül, akkor az RPM automatikusan visszavonja a hibás tranzakciót, kivéve persze amellyik a *%post* parancsfájl futtatása során okozott hibát.

Összefoglalás

A tranzakció alapú RPM visszavonás hatékony módszer az RPM frissítések előtti helyzet visszaállítására. Egy biztos alapot ad, melyre építve a rendszerfrissítő programok (*update*, *yum* és az *apt-get*) automatikus visszavonási képességet nyújthatnak a felhasználónak. Ugyanakkor a tranzakció alapú visszavonást nem igényli mindenki. *Jeff Johnson* szavaival élve „a *--rollback* kapcsoló... tökéletes rendszeradminisztrációt igényel és inkább gépezet mint rendszabály.” A tranzakció alapú visszavonás mindent-vagy-semmit jellegű dolog. Figyelni kell rá, hogy minden törlés újracsomagoldjék, hiszen az RPM tranzakció visszavonási képessége csak annyira használható amennyire az újracsomagolt csomagkészlet. A rendszergazdának biztosítania kell némi külön tárhelyet az újracsomagolt csomagok számára. Végül, az tranzakció alapú RPM visszavonás folyamatos fejlesztés alatt áll. Amennyiben szeretnénk, hogy az RPM rendszerfrissítéseinket gyorsan vissza tudjuk vonni, valószínűleg pontosan ez az amit a doki ajánl.

Linux Journal 2004. május, 121. szám



James Olin Oden (joden@lee.k12.nc.us) programozó a Tekelec-nél. UNIX-típusú rendszereket kezel és alattuk fejlesztett több mint egy évtizede. Ő a Tech Tracker (tt.lee.k12.nc.us) e Web alapú IT-követő rendszer alkotója is.

© Kiskapu Kft. Minden jog fenntartva

Szavazz a CD-mellékletéről!

Tavasszal „Szerkeszd te is a Linuxvilágot!” felhívással egy on-line kérdőív kitöltésére kértük olvasóinkat honlapunkon, melynek értékelése a júliusi számban jelent meg (a bővebb változat honlapunkon is elérhető <http://www.linuxvilag.hu/hir/1022/711.html>). Az eredmény alapján készítettünk egy tervezetet a CD-melléletekre vonatkozó változtatásokra. Ennek megvalósításáról a Ti szavazataitok fognak dönteni, ezért kérünk mindenkit, hogy válaszoljon 3 kérdésre ezen az oldalon:

http://www.linuxvilag.hu/kerdoiv_cd

