

## A Bricolage

A Salon és más népszerű webhelyek mögött meghúzódó tartalomkezelő rendszer egyszerűen használható saját webhelyünk írói és szerkesztői számára is.

**A**z elmúlt néhány hónapban bemutattunk egy pár Zope alapú tartalomkezelő rendszert. Természetesen nem a Zope az egyetlen versenyző a nyílt forrású tartalomkezelő rendszerek mezőnyében. Az egyik egyre jelentősebbé váló csomag a Bricolage, amelyet *David Wheeler* írt és tart karban, és amely a `mod_perl` csomagon és a PostgreSQL adatbázis-kezelőn alapul.

A Bricolage program használatához nem szükséges számítástechnikai ismeret. Igaz, hogy a program módosítása, illetve karbantartása nagy szakértelmet kíván, de az Apache webkiszolgálóval vagy a Perl nyelvvel dolgozó emberek általában programozók vagy rendszergazdák, míg a Bricolage felhasználói a webhely írói, szerkesztői és készítői.

A Bricolage a való világban is bizonyított. Az Apache és a Perl évekig küzdött azért, hogy bekerüljön a főáramba. A Bricolage már jó ideje a Salon magazin tartalomkezelő rendszere, és az eWeek, valamint a Register webhelyeknél is folyamatban van az átállítás. Ráadásul a szakma képviselői közül, akik nem mindig nyilatkoznak kedvezően a nyílt forrású programokról, egyre többen próbálják ki és írnak bírálatot a Bricolage csomagról. A többség kitűnő programnak értékeli, amely felveszi a versenyt sok ezer dollárba kerülő, zárt kódú vetélytársaival. Ebben a hónapban áttekintjük a Bricolage telepítését és használatba vételét. A következő pár hónapban sok szempontból megvizsgáljuk a programot, kitérve a különféle webhelytípusokhoz és a közléshez való testreszabásra.

### Alapok

Minden tartalomkezelő rendszer magja egy adatbázis. A kereskedelmi tartalomkezelő programok általában az Oracle vagy a Microsoft SQL Server adatbázis-kezelőt használják. Számos nyílt forrású program, beleértve több PHP alapú tartalomkezelő rendszert is, a MySQL adatbázis-kezelőt használja. Ezzel szemben a Bricolage a PostgreSQL-t választotta háttértárolóként. A PostgreSQL-t gyakran úgy emlegetik, mint „a másik” nyílt forrású adatbázis-kezelőt, pedig már régóta támogatja a tranzakciókezelést, lehetővé téve, hogy több lekérdezést vagy parancsot egy kötegbe fogjunk, és „mindent vagy semmit” alapon hajtunk végre. A PostgreSQL olyan képességekkel is rendelkezik, amit a komoly adatbázis-felhasználók elvárnak, például nézetek, felhasználó által megadott függvények, rész kiválasztások, egyesítések, idegen kulcsok és sértetlenség-ellenőrzés. A PostgreSQL támogatja a Unicode szabványt, ami egyre inkább fontos a többnyelvű webhelyek kezelésénél. A Bricolage a PostgreSQL-t használja háttértárolóként, de magát az alkalmazást Perlben írták. Legalább két módszer ismeretes a kiszolgálóoldali webes Perl-programok futtatására. Az egyik a CGI, ami lassú, de biztonságos és hordozható, a másik a `mod_perl`, ami gyors, de a biztonsággal gondok lehetnek, és csak az Apache webkiszolgálóval működik. A Bricolage a `mod_perl` segítségével dolgozik, ami azt jelenti, hogy a kódja – Perl-modulok rendszere – csak egyszer fordul le, „Perl-kódok” formájában a memóriában tárolódik, és onnan

hajtódik végre minden alkalommal. Ez a megoldás azt eredményezi, hogy a Bricolage működése gyors lesz. Mint már említettem, a `mod_perl` csak az Apache webkiszolgálóval működik. Bár folyamatosan dolgoznak a `mod_perl` átvitelén az Apache 2.x számára, a cikk írásának idején (2003 júniusa) még csak az Apache 1.x változatával működik együtt. Mivel az Apache 1.x több különálló folyamatot futtat, ahelyett hogy egy folyamaton belül több szálát indítana, nincs lehetőség arra, hogy igazi adatbázisgyűjtőt hozzunk létre a különféle gyermek HTTP-kiszolgálók számára. A létrehozott adatbázis-kapcsolat viszont életben tartható az Apache és a PostgreSQL között az Apache: :DBI modul használatával. A Bricolage pontosan ezt teszi, így az adatbázis-kapcsolatot nem kell minden alkalommal újra létrehozni, amikor egy felhasználó lekér valamit. A Bricolage az adatokat Perl/HTML-sablonok segítségével jeleníti meg a felhasználóknak. Seregnyi ilyen sablon van általában is a Perlhez, a `mod_perl`-hez pedig különösen sok. Régóta nagy rajongója vagyok a HTML: :Mason modulnak, és az egyik ok, amiért ennyire lelkesedem a Bricolage programért, az az, hogy a kedvenc megoldásaimat – PostgreSQL, `mod_perl`, Apache és HTML: :Mason – egy, a végfelhasználó számára jó alkalmazásba fogja össze.

### Telepítés

A Bricolage telepítése nem egyszerű folyamat. Ez nem a Bricolage szerzőinek és karbantartóinak hibája, inkább azért ilyen, mert a Bricolage rengeteg modult használ CPAN-ból (Comprehensive Perl Archive Network). Még ma sem megy olyan simán a telepítés, ahogy mennie kellene, de a dolog folyamatosan fejlődik, és az újabb változatokat egyre könnyebb telepíteni.

A legegyszerűbben úgy telepíthetjük a Bricolage programot, hogy miután a Perl, az Apache és a `mod_perl` – statikus modulként, nem Apache dinamikus megosztott objektumként (DSO) – telepítve van, a `Bundle: :Bricolage` virtuális modult telepítjük a CPAN-ból. A Perl-modulokat úgy lehet telepíteni a CPAN-ból, hogy elindítjuk a CPAN-héjat a `perl -MCPAN -e 'shell'` paranccsal, és a parancssorban kiadjuk az `install Bundle: :Bricolage` parancsot. Ha viszonylag új Perl-héjat használunk, és megadtuk a `PGINCLUDE` és `PGLIB` környezeti változókat, akkor minden modulnak a legújabb változata magától és gond nélkül letöltődik és telepítődik. A folyamat hosszú és bonyolult lehet, ha hozzájár hasonlóan belebonyolódasz a CPAN függőségeinek erdejébe. Nem minden CPAN-csomag adja meg világosan, hogy milyen más csomagoktól függ. Először az LWP-t és a `Bundle: :CPAN`-t telepítettem a CPAN-héjból. Ezután megkíséreltem a `Bundle: :Bricolage` telepítését (Red Hat 7.3-as rendszeren), de nem sikerült elsőre telepíteni a `Cache: :Cache` modult, másodsorra viszont igen. A `DB_File` telepítése is sikertelen volt, mert nem volt telepítve a `db3-devel` RPM-csomag, és ez gondot okozott az Apache: :Session telepítésénél, amely azután a HTML: :Mason telepítését tette lehetővé, amitől a Bricolage

függ. Gondok voltak a *libapreq* telepítésével is (mert az Apache már futott ugyanazon a kapun), és az `XML: :Parser` csomaggal is (mert az `expat-devel` RPM-csomag nem volt telepítve). Szerencsére egy CPAN-köteg telepítése után a rendszer kiírja, hogy mely csomagokat nem sikerült tisztán telepíteni. Megkísérrelhetjük ismét a köteg telepítését (ekkor a CPAN közli, hogy mely modulok vannak már telepítve, és melyek még nem). A `Bundle: :Bricolage` nem a Bricolage moduljait telepíti, hanem csak azokat, amelyekről a Bricolage függ. Ha meggyőződünk arról, hogy a `Bundle: :Bricolage` telepítése sikeres volt, akkor töltsük le a legújabb Bricolage-tarcsomagot a Bricolage honlapjáról (☞ <http://www.bricolage.cc>), csomagoljuk ki, majd adjuk ki a `make` parancsot. A `Makefile` ellenőrzi, hogy a kötelező és választható Perl-csomagok közül melyek vannak telepítve, és megkérdezi, hogy telepíteni szeretnénk-e a hiányzókat. Ellenőrzi azt is, hogy a `mod_perl` statikus modulként (és nem DSO-ként) van-e fordítva, és hogy a PostgreSQL telepítve van-e. Végül néhány felhasználói nevet és jelszót kér, amelyek a Bricolage adatbázisának elkészítéséhez szükségesek, valamint telepíti a rendszerre a program `HTML: :Mason` összetevőit. A kérdések megválaszolása után a telepítést a következő parancs kiadásával indíthatjuk:

```
make cpan && make test && make install
```

Ez a parancs letölti és telepíti a szükséges modulokat a CPAN-ból, ellenőrzi a Bricolage telepítését, hogy minden rendben van-e, és ha igen, telepíti magát a Bricolage programot.

A Bricolage telepítése közben felmerülhetnek gondok, de az egész telepítés meglepően könnyű a rendszer bonyolultságához képest. A `Makefile` értelmes, előre kitöltött válaszokat javasol, kényszerít a „bric” PostgreSQL-felhasználó jelszavának megadására, és általában gondoskodik arról, hogy minden jól és könnyedén menjen.

A telepítés után kiadtam a `bric_apachectl` parancsot az Apache elindításához, a webböngészőmet a gyökérkönyvtárra állítottam, és megjelent az üdvözlő bejelentkezési képernyő. A telepítés sikerült! A Bricolage leírása javasolja, hogy azonnal változtassuk meg a rendszergazda jelszavát, ami ennek megfelelően *change me now!* (változtass meg!) értékre van alapból beállítva. Itt említendő meg, hogy a Bricolage elég sokat használja az előugró ablakokat. Én személy szerint utálok ezeket ablakokat, még akkor is, ha nem hirdetés célból jelennek meg. Bár megértem, hogy a HTML-ben tervezett felhasználói felületek gyakran nem nélkülözhetik az előugró ablakok használatát, nem bánám, ha a Bricolage kevesebbet használná ezt a megoldást. További zavaró tényező, hogy sok belső hivatkozás JavaScript-függő, így nem használhatom a kedvenc böngészőmben a középső egérgombot, hogy ne új ablakban, hanem új böngészőablakon jelenjen meg az oldal. A fenti apróságok nem sokat változtatnak azon a tényen, hogy a Bricolage igazán fantasztikus csomag.

### A motorháztető alatt

A Bricolage több részből tevődik össze: adatmodell a PostgreSQL-ben, számos Perl-modul és sablon, amelyek a modulok által megszerzett adatokat jelenítik meg a felhasználók számára. Ez hasonló más, bonyolult adatbázis-hátterű rendszerekhez. Bár a Bricolage kevesebb feladatot kíván megoldani, mint az OpenACS, és nyilvánvalóan más módszereket használ célja elérésére, mégis nagyon hasonlít rá és azokra a rendszerekre, amelyek háromrétegű felépítést használnak. Ha belenézünk a Bricolage csomag telepítőkönyvtárába, lát-

hatjuk a rendszer által használt PostgreSQL adatbázis-meghatározásokat. A kezdő PostgreSQL-felhasználók meglepődhetnek azon, amit ebben az állományban látnak, például sorszámokat és feltételeket. A sorszám (sequence) egy különleges számobjektum a PostgreSQL-ben, amelynek az értékeit sosem lehet újra felhasználni. Leggyakrabban arra használják őket, hogy egyedi azonosítókat hozzanak létre egy rendszerben, különösen, amikor elsődleges kulcsként is felhasználják a programozók. Ha a PostgreSQL-ben egy oszlopot `SERIAL` típusuként határozunk meg az `INTEGER` helyett, akkor tulajdonképpen sorszámot hozunk létre. A PostgreSQL régebbi változataiban kényelmetlen volt a `SERIAL` oszlopot tartalmazó táblák törlése, először a táblát kellett törölni, azután a sorszámot, amely a `SERIAL` oszlophoz volt rendelve. A PostgreSQL 7.3-as megjelenésétől kezdve a tábla törlése magával vonja a `SERIAL` oszlopokhoz rendelt sorszámok törlését is. A feltételek (constraint) lehetővé teszik, hogy az adatbázis visszautasítsa az olyan `INSERT` és `UPDATE` parancsokat, amelyek egy megadott tartományon kívül eső értéket próbálnak beállítani. Például a Bricolage megad egy media táblát, amelyben minden elem 1-től 5-ig terjedő fontossággal (priority) rendelkezik, az alapértelmezett érték 3. Az oszlop meghatározása így néz ki:

```
oids priority NUMERIC(1,0) NOT NULL
      DEFAULT 3
      CONSTRAINT ck_media__priority
      CHECK (priority BETWEEN 1 AND 5)
```

A feltételnek adhatunk nevet, ebben az esetben ez a név `ck_media_priority`. A névadás egyszerűbbé teszi a hibák megtalálását és javítását, ugyanis amikor ebbe az oszlopba érvénytelen értéket próbálunk beszúrni, a PostgreSQL kijelzi a megsértett feltétel nevét. Ez eléggé nagy segítség akkor, amikor az adatbázis-meghatározásban vagy az adatbázissal együttműködő alkalmazásban kell a hibát megtalálni. Az is meglepő lehet, hogy kevés függvény lett meghatározva.

## KAPCSOLÓDÓ GÍMEK

A legfőbb ismeretforrás a Bricolage honlapja:

☞ <http://www.bricolage.cc>. Ez az oldal tartalmaz hivatkozásokat a letölthető forráskódra (amely a ☞ <http://SourceForge-on> van), a leírásra és a hibajelentéseket és továbbfejlesztési kéréseket gyűjtő Bugzilla-rendszerre ([bugzilla.bricolage.cc](http://bugzilla.bricolage.cc)).

Sok Bricolage programmal foglalkozó levelezőlista van a SourceForge-on a ☞ <http://sourceforge.net/projects/bricolage> weblapon.

A `mod_perl` honlapja a ☞ <http://perl.apache.org>

A Mason-könyv honlapja

(☞ <http://www.masonbook.com>) és a Mason honlapján (☞ <http://www.masonhq.com>) tudhatunk meg többet.

*David Wheeler*, a Bricolage főprogramozója és karbantartója szintén fenntart egy honlapot, amelynek címe ☞ <http://david.wheeler.net>. Vállalkozása, a Kineticcode a ☞ <http://www.kineticcode.com> címen érhető el.

Az Apache webkiszolgáló ☞ <http://httpd.apache.org>

A PostgreSQL lehetővé teszi, hogy függvényeket adjunk meg, amelyeket többféle programnyelv – például szabványos SQL, eljárás alapú PL/PGSQL, adatbázisokra kihegyezett Perl-változat, Python vagy Tcl – segítségével valósíthatunk meg. Természetesen a Bricolage adatmodelljének magját a táblák képezik. A *person* tábla tartalmazza a rendszer felhasználóit, az *org* tábla írja le a szervezeteket, és a *person\_org* tábla kezeli az előző két tábla metszetét.

A Bricolage belsejének működését könnyen megérthetjük, ha tanulmányozzuk az adatbázis-meghatározásokat. Például egy új cikket – egy áradozó kritikát a Core Perlről – adok hozzá Bricolage rendszerhez az alapértelmezett beállításokkal, azaz az alapbeállításokat használó rendszergazda nevében. A művelet hatására a *story* táblába beszűrődött egy új rekord, hozzá lett rendelve a fontosság, a megjelenés és a lejárat dátuma, a változatszám (ugyanis a Bricolage a cikkek változatait is kezeli), és annak jelzése, hogy a cikkváltozat megjelent-e.

Számos kulcs – amelyek idegen kulcsok más táblákhoz – jelzi, hogyan illeszkedik bele az adott cikk a rendszerbe. Láthatjuk, hogy a rendszergazda hozta létre, mert az *usr* oszlop az *usr* táblára mutat; része a *story* munkafolyamatnak, ami elkülönül a többi megadott munkafolyamattól, és a *workflow\_id* oszlopban hivatkozik rá egy külső kulcs. Ez az *edit* munkasztáról érkezett, amely megkülönböztethető a többi munkasztaltól (például *legal* és *publishing*), és az *edit* külső kulccsal hivatkozik a *desk* táblára; és ez egy *book review* (könyvkritika) típusú cikk, mert az *element\_id* oszlop az *element* táblára mutat. Ezek a táblák is össze vannak kötve más táblákkal, így további kiegészítő adatokat (például formátum) is tárolhatunk.

Röviden: a *story* tábla áll a Bricolage adatmodelljének központ-

jában, ahogy az logikus is egy olyan tartalomkezelő rendszer-nél, aminek a tartalom köré kell szerveződnie.

Nagyon ajánljuk a Bricolage programot mindazoknak, akik kezdők a relációs adatbázisok világában, és egy olyan nyílt forrású programot szeretnének tanulmányozni, amely kifinomult módon használja őket.

## Összegzés

Ebben a hónapban megismerkedtünk a Bricolage programmal, amely egy nyílt forrású, *mod\_perl*-en és PostgreSQL-en alapuló tartalomkezelő rendszer. Megtanultuk a telepítését és a használatba vételét, aztán kicsit beleástuk magunkat az adatmodellbe, amelyet a Bricolage a cikkek elemeinek a nyilvántartására használ. A következő hónapban megnézzük, hogyan kell elemeket, kategóriákat és médiatípusokat megadni, ami lehetővé teszi, hogy ne csak a rendszert piszkáljuk, hanem tényleges tartalmat tehessünk ki a webhelyünkre. Ez után még jobban elmerülünk majd a rendszerben, megvizsgáljuk a Mason-sablonokat, amelyeknek a segítségével olyanra változtatható a Bricolage alapértelmezett kinézete, ami közelebb áll a saját webhelyünkön megjeleníteni kívánt látványtervhez.

*Linux Journal* 2003. szeptember, 113. szám



**Reuven M. Lerner** (☞ <http://www.lerner.co.il/atf>)

Nyílt forrású programokra, valamint web- és adatbázis-alkalmazásokra szakosodott tanácsadó.

Könyve, a *Core Perl*, 2002 januárjában jelent meg a Prentice Hall gondozásában. Reuven feleségével és lányaival Izraelben, Modi'in-ben él.

