

Tartalomkezelés

Kövessük nyomon honlapunk frissítését a Mason segítségével!

Társlapunk az utóbbi néhány hónapban a Masonnal, *Jonathan Swartz* webfejlesztő rendszerével foglalkozott, mely az Apache-t, a **mod-perl**-t és a HTML/Perl mintákat kapcsolja össze.

Általában a Mason a webalkalmazásokhoz kötődik – elsősorban azokhoz, amelyek háttéradatbázist alkalmaznak, és dinamikus tartalmuk is van. A Mason-t én rendkívül jól használható eszköznek találtam a tanácsadói feladataimban, mivel lehetővé tette a weblapok gyors és könnyű előállítását.

A csomag azonban egyszerű kiszolgálóoldali webalkalmazásoknál többre is használható. Az egyik legérdekesebb Mason-alkalmazás a Mason Content Manager (Mason-CM), mely alapvető tartalomkezelési lehetőségeket teremt, a fájlok külső kiszolgálóra (production server) történő felvitelétől kezdve, a beépített helyesírás-ellenőrzőn át az RCS változatkezelésig, valamint a Mason összetevőinek szerkesztéséig.

Mivel a Mason-CM Masonben íródott (tehát szükséges hozzá az Apache, a mod_perl és a megfelelő Perl modulok), természetesen jól működik egyszerű, csak HTML-t és képeket tartalmazó statikus lapokkal is.

Mindenképpen ajánlom, hogy még azok a webmesterek is vessenek egy pillantást a Mason tartalomkezelőjére, akiket egyébként nem érdekel a Mason és a mod_perl, hiszen ennek az egyszerű és ingyenes csomagnak oly sok hasznos tulajdonsága van.

Mi az a tartalomkezelés?

Ahogy a webhelyek egyre bonyolultabbak lesznek, úgy válnak a hozzájuk tartozó szervezési munkák is egyre összetettebbé. Míg régebben általános volt, hogy egy profi webhelyet egyetlen ember kezelhetett, ma már nem ritka, hogy akár egy kisebb vagy közepes méretű helyet is legalább három ember tart karban: a tartalomfelelős vagy szerkesztő, a tervezőgrafikus és a programozó. Még az ilyen kis csapatoknál is előfordul, hogy több ember próbálja ugyanazt a fájlt módosítani egy időben. Ezt a gondot már évekkel ezelőtt megoldották az olyan változatkezelő rendszerek segítségével, mint amilyen az RCS vagy a CVS. Csakhogy ezeket a rendszereket programozók tervezték programozóknak, így gyakran elég elrettentők lehetnek egy tervezőgrafikus vagy szerkesztő számára.

A Web számos olyan új kihívást is jelenthet, amely különbözik a hagyományos programozás világában megszokott fejlesztésektől. Például a programokat általában megírják, lefordítják, ellenőrzik, és hibakeresést végeznek rajtuk, majd az egész kezdődik elölről, míg végül kiadják. A Web ellenben másként működik. Amint a HTML-fájl a kiszolgálóra kerül, azonnal hozzáférhető mindenki számára az Interneten. Ez jó, hiszen az esetleges hibákat azonnal felfedezhetjük, de azt is jelenti, hogy a helyek szabályos időközönként frissíthetik tartalmukat, hosszan tartó eljárás igénybevétele nélkül.

Egyúttal persze a módosítás eredménye – legyen az oka fejlesztés vagy akár hiba – azonnal elérhető mindenki számára, aki éppen a jó URL-t írja be a rossz időben. Emiatt a legtöbb közepes, illetve nagy webhelyet két webkiszolgálón futtatják.

A belső kiszolgálót (staging server) a fejlesztők, a szerkesztők és a grafikusok írhatják, szerkeszthetik, majd ellenőrizhetik változtatásaik hatását. Csak akkor kerülhetnek át az adatok a második kiszolgálóra, ha minden fájl rendben van. Ezt a helyet külső kiszolgálónak (production server) nevezik.

Természetesen ez a két kiszolgáló futtat ugyanazon a számítógépen is. A lényeg, hogy valahogyan elválasszuk őket egymástól, hogy a látogatók ne férhessenek hozzá a fejlesztés alatt álló tartalomhoz. Azt is lehetővé kell tennünk, hogy a belső és a külső kiszolgálónak azonos könyvtárszerkezete legyen.

A Mason-CM a Mason alkotóelemeiből épül fel, s viszonylag egyszerűvé teszi bárki számára egy tartalomkezelő rendszer beindítását a saját rendszerén. A felhasználói felület nem éppen csodaszép, és néhány kisebb nehézséggel találkoztam a telepítés közben. Mindazonáltal feladatát megfelelően ellátja, és lehetővé teszi, hogy egyszerre több felhasználó is dolgozzon ugyanazon a feladaton anélkül, hogy egymás „tyúkszemére lépneek”.

A Mason-CM telepítése

A Mason-CM telepítése előtt először szükség lesz egy működő Apache-változatra, **mod_perl** és Mason támogatással. Ezenkívül le kell töltenünk az alábbi Perl modulokat a CPAN-ról: MLDBM, Image::Size, URI::Escape és File::PathConvert. Győződjünk meg arról, hogy ezek a PerlModule jelzéssel kerültek be az Apache beállításfájljába (httpd.conf), és növeljük meg a különböző Apache al-folyamatok (child process) számára adott megosztott memória méretét. A Mason-CM gzippel tömörített tar fájlként tölthető le a termék honlapjáról (lásd a cikk végén a kapcsolódó címeket). A tárolófájl a Mason összetevőket tartalmazó könyvtárba érdemes kicsomagolni. A magam részéről a /usr/local/apache/mason/cm könyvtár választottam. Ha az állományt kicsomagoltuk, a /cm könyvtárnak egy README és egy INSTALL fájl kell tartalmaznia. Bár itt a telepítéshez szükséges összes lépést ismertetem, mégis jó ötletnek tűnik végigolvasni ezeket a fájlokat, csak a biztonság kedvéért. Mivel a Mason-CM-nek kell lehetővé tennie azt, hogy minden fájl csak egy felhasználó módosíthasson egyszerre, valamint azért hogy a tartalomkezelő rendszert csak az arra hivatott felhasználók érhesék el, a /cm könyvtár eléréséhez HTTP-hitelesítést kell beállítanunk (a Mason-CM csak jelszóval védett könyvtárból futtatható). Egy könyvtárat a legegyszerűbben úgy tehetünk jelszóvédté, hogy egy **.htaccess** fájlt helyezünk el benne. A .htaccess fájl felülbírálja az alapértelmezett Apache beállításokat az adott könyvtárra, illetve annak összes alkönyvtárára. Itt van például a saját tartalomkezelőmhöz tartozó .htaccess fájl:

```
AuthName "Content Management System"
AuthType Basic
AuthUserFile /usr/local/apache/conf/st-passwords
require valid-user
```

Az AuthName érték határozza meg a jelszó bekérésekor megjelenő szöveget. (Enélkül a felhasználó nem tudná, hogy éppen melyik rendszer kér felhasználói nevet és jelszót.) Az AuthUserFile egy, a felhasználói neveket és a kódolt jelszavakat tartalmazó fájlra mutat. A jelszófájl a honlap gyökérkönyvtárán kívül is elhelyezkedhet, így az böngészőn keresztül elérhetetlen. A jelszófájl elkészítéséhez vagy szerkesztéséhez használjuk a htpasswd programot. Ez alapértelmezés szerint a /usr/local/apache/bin-be kerül. A require valid-user hatására a felhasználói név, vagy az AuthUserFile-ban

található jelszó nélkül a felhasználó nem érheti el a könyvtárat.

Ha a **.htaccess** fájlban semmi hatása nem lenne, ellenőrizzük a `httpd.conf` fájlban lévő `AllowOverride` értéket. Ugyanis ez teszi lehetővé, hogy az Apache beállításait a `.htaccess` felülbírálhassa. Alapértelmezés szerint az Apache-kiszolgálók nem engedik meg a `.htaccess` fájlokban az `AuthConfig` típusú módosítók használatát. Ezt megváltoztathatjuk, ha az alábbi sorokat a `httpd.conf`-ba írjuk:

```
<Directory /usr/local/apache/mason/cm>
AllowOverride AuthConfig
</Directory>
```

Egy kisebb hibát találtam a telepített Mason-CM változatomban. A legtöbb összetevő `.html` kiterjesztés nélküli, ez nehézkessé vagy akár lehetetlenné is teheti az Apache számára a tartalom `text/html`-ként való azonosítását. Így, bár a Mason-CM összetevők HTML-formátumú kimenetet készítenek, a tartalmat a böngésző formátumát ASCII szövegnek ismerte fel. Tehát az Apache kis segítségére szorult. A `mod_perl content_type` eljárásával közvetlenül beállítottam a tartalom típusát. A következőket a `/cm` könyvtárba helyezve, a könyvtár összes dokumentumát automatikusan `text/html` típusúra állíthatjuk be:

```
<% $m->call_next %>

<%init>
$r->content_type("text/html");
</%init>
```

Az én `mason.pl` beállításfájlom figyelmen kívül hagy mindent, ami nem szövegfájl, így biztos lehetek benne, hogy a fentiek nem fognak véletlenségből JPEG és PNG képeket `text/html` típusúként felismerni.

Beállítás

A Mason-CM most már a helyére került, a működéshez azonban az Apache-nak néhány Perl modult be kell töltenie. A Mason beállításfájlba (amit én `mason.pl`-nek neveztem el, de a Mason leírása `handler.pl`-ként hivatkozik rá) szúrjuk be a következő Perl kódot:

```
@cx: {
    package HTML::Mason::Commands;
    use Fcntl;
    use MLDBM;
    use Image::Size;
    use URI::Escape;
    use File::PathConvert;
    use File::Copy;
    use File::Find;
    use IO::Handle;
    use IPC::Open2;
}
```

Miután az Apache és a Mason már tudja, hogy hol keresse a Mason-CM részeit, itt az ideje, hogy magát a Mason-CM-et is beállítsuk. Majdnem minden beállítást a **cmConfig** módosításával végezhetünk, mely a `/cm` könyvtárban található. A jelenlegi `cmConfig` fájl még a régebbi változatú Mason csatolóval készült; ez kissé zavaró lehet azok számára, akik a Mason 0.80-as változatát szokták meg. Például az indítórész neve `<%perl_init>`, nem pedig egyszerűen `<%init>`, és az egyik összetevő a másikat az **mc_comp** segítségével hívja, a `$m->comp` helyett. Az alkotórészek viszonylag könnyen felismerhetők és megérthetők bárki számára, aki használta már a Masont.

A `cmConfig` fájl elején beállítandó két legfontosabb változó a `$CM_HOME` és a `$CM_DATA`. Ezeket az eredeti `cmConfig` 25. sora határozza meg (ez a `<%perl_init>` rész elején található). Az első a Mason-CM telepítési könyvtárára, a második pedig az általa kezelt adatokat (zárolás, változatszám) tároló könyvtárra hivatkozik. Én így határoztam meg ezeket:

```
$CM_HOME = '/usr/local/apache/mason/cm';
$CM_DATA = '/usr/local/apache/cmdata';
```

A Mason-CM működéséhez mindkét könyvtárnak léteznie kell. Míg a `$CM_HOME` alapértelmezés szerint van beállítva (hiszen várhatóan a `cmContent` a `$CM_HOME` belsejében található), a `$CM_DATA` könyvtárat külön létre kell hoznunk.

Jegyezzük meg, hogy ez a könyvtár nem egyezik meg a Mason adatkönyvtárával – ez utóbbi általában a `/usr/local/apache/masondata`. A `$CM_HOME` és a `$CM_DATA` meghatározása után a **%cm_config** beállítására szolgáló rész következik. Itt változókat és azok értékeit találjuk. A legtöbb esetben az alapértelmezett beállítások megfelelőek, most csak azokat tárgyaljuk, amelyeket érdemes vagy szükséges módosítani.

Az admin kulcs a Mason-CM rendszergazdájának levélcímét tartalmazza. Ő felelős a törölt fájlok visszaállításáért, a zárolt fájlok feloldásáért és általában a tartalomkezelő vezérléséért. Alapértelmezés szerint ez `cm-admin-ra` van állítva, de bármit beállíthatunk.

Ágak meghatározása

A branches kulcshoz rendelt érték egy tömbhivatkozás, mely a változatkezelő rendszer *ágait* (branches) írja le. Az összes ágnak a `$CM_HOME` könyvtárban kell elhelyezkednie, így különbséget tehetünk az *alhelyek* (subsites) között. Például egy újság cikkei ágakra bonthatók, külön részt fenntartva a hírek, a sport és az üzleti témakörök számára. Minden ágat egy név azonosít, ezt pedig az ág tulajdonságai követik. Ha honlapunk egy ágat tartalmaz, melynek neve „Elso”, a branches kulcs valahogy így néz ki:

```
branches => [
    Elso => {
        path =>
        '/usr/local/apache/htdocs/staging/content',
        trg_from => 'staging',
        trg_to => 'production',
        components => 0
    }
]
```

A fenti ág a Mason-CM ágválasztójában (branch selector) is megjelenik, „Elso” néven és az `/usr/local/apache/htdocs/staging/content` alatti dokumentumokat irányítja. Győződjünk meg arról, hogy a könyvtár nem / karakterrel végződik, mert ebben az esetben a Mason-CM biztonsági hibával leáll.

A `trg_from` és a `trg_to` kulcsokat egyszerű behelyettesítésre használhatjuk; segítségével itt azt jelezzük, hogy a „staging” karakterláncot „production”-ra cseréljük le, és így másoljuk át a dokumentumokat a belső kiszolgálóról a külsőre. (A Mason-CM ezt a folyamatot „triggering”-nek nevezi.)

Így a tartalom kezdetben az `/usr/local/apache/htdocs/staging/content` könyvtárban van, majd az `/usr/local/apache/htdocs/production/content` könyvtárba kerül. Végül azt is jelezzük, hogy az ág HTML-kódot tartalmaz (és nem Mason-összetevőket): ehhez a `components` kulcsot kell 0-ra állítanunk.

Egy összetettebb webhelyen a következőképpen állíthatnánk be az ágakat:

```
branches => [
  News => {
    path => '/usr/local/ \
apache/htdocs/staging/news',
    trg_from => 'staging',
    trg_to => 'production',
    components => 1,
    hidden => 1
  },

  Business => {
    path => '/usr/local \
/apache/htdocs/staging/business',
    trg_from => 'staging',
    trg_to => 'production',
    components => 1,
    obj_dir => \
'/usr/local/apache/staging/obj',
    hidden => 1
  }
]
```

A fenti Mason-CM beállításnak két ága van, a News és a Business. Mivel a branches egy tömb, ezért az elemeket az eredeti sorrendjükben tárolja. Ez azt jelenti, hogy az ágválasztó a beírás sorrendjében jeleníti meg a fenti ágakat. Az ágak megjelenési sorrendjének módosításához cseréljük fel az ágak tömbhivatkozásait.

Ha az ágakat a fentiek alapján készítettük el, akkor az Apache beállításait átírhatjuk úgy, hogy minden /news-zal kezdődő címet /production/news-ra módosítson:

```
Alias /news
/usr/local/apache/htdocs/production/news
```

Így a belső (staging) kiszolgáló nem érhető el a böngészők segítségével. A webkiszolgálót azonban úgy is beállíthatjuk, hogy minden, a 8080-as (vagy bármilyen más) kapura érkező kérelem a belső kiszolgálóhoz kerüljön.

A hidden érték azt határozza meg, hogy alapértelmezés szerint az ág megjelenjen-e az ágválasztóban. Alap esetben minden ág megjelenik, és azok minden felhasználó számára elérhetők. Az ágak listáját bárki testre szabhatja a Mason-CM kezdőlap jobb felső sarkában található my.CM hivatkozás használatával, illetve bármelyik felhasználó hozzáadhat ágakat a menühöz és törölhet is abból. Az ágak elrejtésével az új felhasználók viszonylag tiszta képet kaphatnak a kezelőrendszer tartalmáról. Az Elso ággal ellentétben a News és a Business ág Mason összetevőket is tartalmaz. Ezek kimácsolása a külső kiszolgálóra abban különbözik a HTML-lapoknál megszokott módszertől, hogy a másolás előtt a Mason-CM megpróbálja lefordítani az összetevőket, és hibakezérést végez azokon. Így aztán egy sérült alkotórész nem okozhatja a külső kiszolgáló leállítását, legfeljebb a belsővel lehetnek bajok. Ha a lefordított Masonösszetevőket egy meghatározott könyvtárban kívánjuk tárolni, az **obj_dir** kulcsra lesz szükségünk.

A cmConfig fájl többféleképpen is átírható a Mason-CM-hez való együttműködéshez, de ha a \$CM_HOME-ot és a \$CM_DATA-t egyszer beállítottuk, máris használhatjuk a Mason-CM-et.

A kezdőlap

A Mason-CM felület főoldalának eléréséhez a böngészőbe a \$CM_HOME-ban megadott címet kell írunk. Nálam ez a `http://localhost/mason/cm/`.

Mivel ez a könyvtár jelszóval védett, ezért a felhasználói névre és a hozzá tartozó jelszóra is szükség lesz. A sikeres belépést követően

a Mason-CM kezdőlapot látjuk, amely körülbelül egy webalapú fájlböngésző. Segítségével az egyes ágak könyvtárai és alkönyvtárai között mozoghatunk, fájlokat nyithatunk meg olvasásra és írásra, illetve név vagy tartalom szerint kereshetünk a fájlok között.

A kezdőlapot mindig könnyen felismerjük: egy bűvész van rajta. Ezt a cmConfig-ban található juggler_src kulccsal tetszőleges képpel helyettesíthetjük. A képre kattintva mindig a Mason-CM kezdőoldalára jutunk vissza.

A kezdőlap jobb oldalán található az ágválasztó, mely a cmConfig-ban meghatározott ágakat sorolja fel. Az ágválasztó hivatkozásaira kattintva az ág külső kiszolgálóra másolását vezérelhetjük. Az éppen szerkesztett ág más háttérrel jelenik meg. Az aktuális könyvtár a képernyő közepén található, current directory fejléccel és alapértelmezés szerint kék háttérrel. Az aktuális könyvtár útvonalának minden összetevője az adott útvonalra mutató hivatkozás, így egyszerűen válthatunk könyvtárat az egérrel. Az alkönyvtárba lépéshez csak annak nevére kell kattintanunk. A képernyő közepén található szövegmező használatával új alkönyvtárt hozhatunk létre.

A current directory (jelenti könyvtár) vonal felett található a keresőrendszer. Nyilván nem én vagyok az egyetlen, aki eddig általában a **grep** vagy a **find** segítségével keresett fájlokat a webhelyen.

A Mason-CM mindkét programot magában foglalja, így még akár a Unixot nem ismerő felhasználóknak is lehetőségük nyílik az ágon belüli keresgélésre. A keresés a Perl szabványos kifejezéseit (regular expressions) is támogatja, tehát a fájlok között sokféle módon kereshetünk: név, de akár tartalom szerint is.

Azért legyünk óvatosak: nem mindegy, hogy mire keresünk rá.

A Mason-CM boldogan végigszalad fájlok ezreire egy-egy összetett szabványos kifejezés beírásakor, nem törődve azzal, hogy a művelet mennyi időt vesz igénybe.

A current directory vonal alatt az adott könyvtárban elérhető fájlok listája található. Minden fájlnál megjelenik a neve, az utolsó módosítás dátuma, az azt végrehajtó felhasználó, valamint a fájl jelenlegi állapota. Ez utóbbi lehet staging (ez azt jelenti, hogy csak a belső kiszolgálón létezik), prod (a fájl ugyanazon változata található mindkét kiszolgálón) és modified (a fájl mindkét kiszolgálón létezik, de megváltozott). A szövegmező és a create gomb használatával új fájl is létrehozhatunk. Ne tévesszük össze az alkönyvtár készítésére szolgáló gombot a fájlkészítő gombbal. Én módosítottam a gombfeliratokat (a dirTable és a fileTable elemeket), hogy a „Könyvtár létrehozása” és a „Fájl létrehozása” feliratok legyenek rajtuk.

Fájlok megtekintése és szerkesztése

A fájl tartalmának megtekintéséhez kattintsunk a nevére a táblázatban. A HTML-forráskód a böngésző felső részében jelenik meg. A lap alsó részén választhatunk a HTML leképezése, a 80. oszlop utáni sortörés (a nyers szöveg megjelenítése helyett), vagy a HTML-sorok számozása közül.

A fájlok szerkesztésére is van lehetőség a Mason-CM-ben, a program kezdetleges, de használható szerkesztőjének igénybevételével. Kattintsunk a fájl név melletti edit hivatkozásra, mire a fájl tartalmát mutató <textarea> elem jelenik meg. A <textarea> mezőbe szöveget gépelve módosíthatjuk a fájl tartalmát. A lap tetején található szövegmező használatával akár másolhatjuk, vagy át is nevezhetjük a fájlt. Ennél a szerkesztőnél egyszerűbbet nemigen készíthettek volna a fejlesztők: a kurzor mozgatását egy alig működő, Emacs-szerű billentyűkezeléssel oldották meg. A legfontosabb, hogy könnyen és gyorsan végezhessünk módosításokat.

A szerkesztőképernyőről számos lehetőség közül választhatunk:

- A save gomb frissíti a fájlt, és visszatér a szerkesztőképernyőhöz.
- A save and exit gomb menti a fájlt és visszatér a Mason-CM kezdőoldalára.

- A save and render gomb leképezi a HTML-fájlt, így a szerkesztett összetevő előképét is megtekinthetjük.
- A képernyő alján található redraw segítségével átméretezhetjük a szerkesztő ablakát.

A Mason-CM zárolja a szerkesztett fájlokat azért, hogy egyszerre csak egy felhasználó férhessen hozzá. A szerkesztett fájl a kezdőlap tetején található vörös téglalapban jelenik meg. Itt csak a megnyitott fájlokra mutató hivatkozások láthatók, ezekre kattintva azonnal szerkeszthetjük a fájlt, illetve az unlock lapon feloldhatjuk a zárolást. Ha egy másik felhasználó által éppen szerkesztett fájlt próbálunk megnyitni, a Mason-CM nem jeleníti meg a szerkesztőt. Ha a fájl zárolását feloldjuk (unlock), a többi felhasználó ismét hozzáférhet.

A fájlok másolása a külső kiszolgálóra

Ha úgy tűnik, hogy a fájl helyesen működik a belső kiszolgálón, akkor átmásolhatjuk a külsőre. Ennek végrehajtásához válasszunk ki egy vagy több fájlt a könyvtárból, a fájltáblákat bal oldalán található jelölőnégyzetekkel. Ezután kattintsunk a lap alján lévő trigger gombra, ekkor a fájlokat a külső kiszolgálóra másoltuk, s így azonnal a webhely friss (current) másolatává válnak.

A könyvtárban található összes fájlt egyszerre frissíthetjük a lap alján, közvetlenül a trigger gomb mellett található check all négyzet bejelölésével. Ez különösen akkor hasznos, ha új könyvtárat hoztunk létre, és egyszerre szeretnénk használni az összes elemet.

Ha egy fájl külső változatát módosítjuk, akkor a másolás fordítva is végbemegy („reverse-trigger”), azaz a belső kiszolgálóról a külsőre kerül. Ez meglehetősen veszélyes művelet, ezért a Mason-CM a művelet engedélyezése előtt rákérdez.

Helyesírás-ellenőrzés

Ha az alapvető Mason-CM szolgáltatások már működnek, valószínűleg néhány további képességet is szívesen kipróbálnánk. Talán a legérdekesebb lehetőség a helyesírás-ellenőrzés. Ez egy Mason összetevő, mely az ispell segítségével ellenőrzi a dokumentum helyesírását (sajnos csak angol nyelven). A Mason helyesírás-ellenőrző figyelmen kívül hagyja a HTML-tagokat, így nem kell például a href és társai szótárba illesztésével bajlódni. A helyesírás-ellenőrző indításához töröljük a megjegyzésjelet az ispell, main_dict és a supp_dict kulcsok sorából a %cm_config-ban:

```
ispell => '/usr/bin/ispell<\#180>',
main_dict => '/usr/lib/ispell/english.hash',
supp_dict => "$CM_DATA/suppDict",
```

Ha ezeket a kulcsokat egyszer meghatároztuk, a Mason-CM szerkesztőben megjelenik egy spell check jelölőnégyzet. Ennek bejelölése után a save, save and render vagy a save and exit használata során a szöveget ellenőrzi. A félregévelt szavakat egy kis JavaScript program segítségével javíthatjuk: választhatunk másik szót a szótárból, figyelmen kívül hagyhatjuk a hibát, vagy felvehetjük a szót a szótárba. A Mason-CM-rendszerben minden felhasználó ugyanazt a szótárat használja, tehát az egyik felhasználó által hozzáadott szavakat mindenki elérheti. Ez a módszer azonban veszélyt is hordoz magában, hiszen bármelyik felhasználóval megtörténhet, hogy véletlenül hibás szót vesz fel a szótárba. Legyünk óvatosak a javaslatokkal kapcsolatban!

Változatkezelés

A Mason-CM a változatkezeléshez szükséges RCS használatát is lehetővé teszi. Ehhez szükség van ahhoz, hogy a mason.pl elkészítse az Rcs modulok Image::Size, URI::Escape, és File::PathConvert részeit. Ezután a következő sorokat a cmConfig-ban meg kell

Kapcsolódó címek

A Mason honlapjáról tölthetjük le a Mason és a Mason-CM legutóbbi változatát. ➔ <http://www.masonhq.com/>.

Két hasznos cikk található a CVS és webes fejlesztés témakörében az alábbi címeken:

➔ <http://durak.org/cvswebsites/> és a

➔ <http://www.daemonnews.org/199903/websites.html>

A Mason-CM futtatásához szükséges összes Perl modult letölthetjük a CPAN (Comprehensive Perl Archive Network) rendszeréről, mely tulajdonképpen FTP-helyek hálózata. Keressük meg a hozzánk legközelebb eső CPAN tükörszolgáltatót a ➔ <http://www.cpan.org/> címen, vagy használjuk a Perl minden változatában megtalálható CPAN.pm-et.

határozni, vagy ki kell törölni előlük a megjegyzésjelet:

```
rscs_bin => "/usr/bin",
rscs_files => "$CM_DATA/archive",
```

Az értékek beállítása után egy version label (változati címke) szövegmező jelenik meg a szerkesztőlap tetején. Ha mentéskor beírjuk a változati címkét, akkor az RCS automatikusan elindul, és a fájl régebbi változatát is megtartja.

A változatkezelés elindításakor a fájllistán a versions címke is megjelenik, ha rákattintunk, akkor a dokumentum történetét láthatjuk, a diff programot használhatjuk grafikus környezetben, és a régebbi változatokat is ellenőrizhetjük. A változatkezelés szinte nélkülözhetetlen a nagyobb webhelyeknél, mivel a hibák bármikor bekövetkezhetnek, és ilyen esetben kényelmesen visszaállíthatjuk a régebbi, de megbízható változatot.

A helyesírás-ellenőrzésen és az RCS-en kívül a Mason-CM jó néhány szolgáltatást tartalmaz még: a felhasználók feltölthetik a fájlokat HTTP-n és FTP-n keresztül, a rendszergazdák könyvtárként korlátozhatják a felhasználói hozzáférést. Mivel a Mason a Perl egy könnyen érthető változatában íródott, egyszerűen bővíthetjük képességeit (HTML-ellenőrzés másolás előtt stb.).

Összefoglalás

A Mason nagy segítségünkre lehet webhelyek kialakításakor, de a Mason-CM bizonyítja igazán az eszköz rugalmasságát. A Mason-CM megmutatja, hogy a Mason összetevőivel olyan eszközöket is készíthetünk, melyek nem közvetlenül a Weben keletkezett tartalomra vannak hatással. A sokféle Mason-CM eszköz változatossága meggyőzött arról, hogy szinte mindent megoldhatunk a csomag segítségével. Bár én valószínűleg ezután is a GNU Emacs szerkesztőt használom, több ügyfelem honlapján fogom alkalmazni a Mason-CM-et. Olyanokén is, akik Masont használnak a tartalom elkészítéséhez, illetve olyanokén is, akik egyszerűbb, kevésbé fejlett eszközökkel oldják meg a feladatot.



Reuven M. Lerner (reuven@lerner.co.il) egy internetes módszerekre szakosodott izraeli tanácsadó cég tulajdonosa. Éppen mostánában fejezi be (végre!) a *Core Perl* című könyvét, mely a Prentice-Hall kiadó gondozásában jelenik majd meg. Az ATF honlapja:

➔ <http://www.lerner.co.il/atf/>.