

# MEO

MAGYAR EGYSÉGES ONTOLÓGIA

<http://ontologia.hu/meo>

**A PROTÉGÉ 3.1.1 SZERVER HASZNÁLATA**

[http://ontologia.hu/document/proj\\_doc/ontologiai\\_szerkesztok/meo\\_protege\\_server\\_guide\\_hun.pdf](http://ontologia.hu/document/proj_doc/ontologiai_szerkesztok/meo_protege_server_guide_hun.pdf)

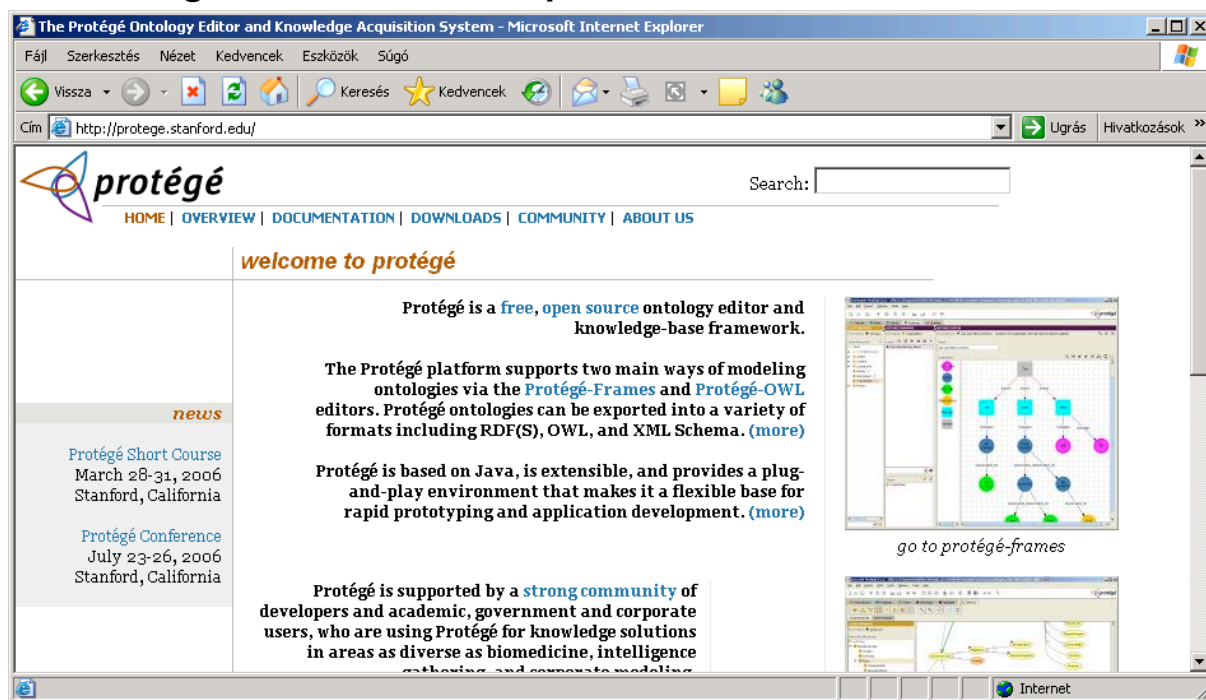
SZEGEDI TUDOMÁNYEGYETEM  
SZÁMÍTÁSTECHNIKAI TANSZÉKCSOPORT  
TÁMOGATÓ

BUDAPEST  
2006.06.03.  
NKFP-2/042/04

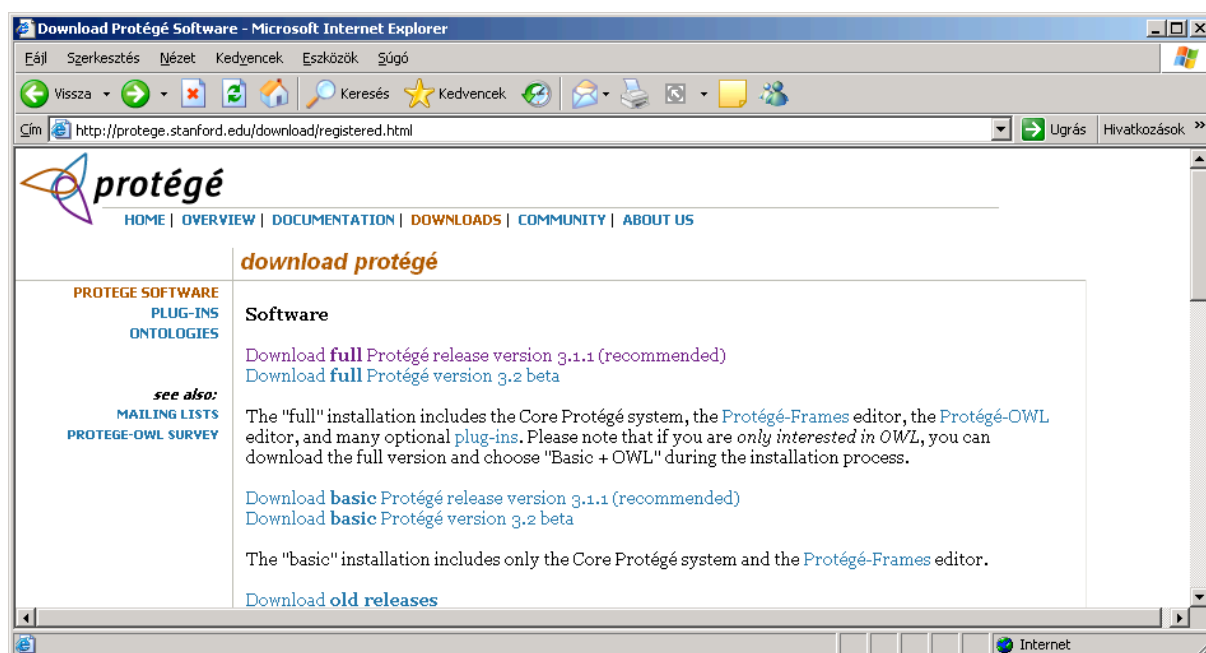
## A Protégé 3.1.1 szerver használata

A MEO (Magyar Egységes Ontológia) NKFP 2004 projekt közös munkát támogató ontológia-szerkesztő környezetet alakított ki. A rendszer alapja a Protégé 3.1.1. általános felhasználói licenccel rendelkező ontologiaszerkesztő program. Az alábbiakban ennek a környezetnek a környezet használatával kapcsolatos tudnivalókat foglaljuk össze.

### 1. A Protégé 3.1.1. szerver telepítése

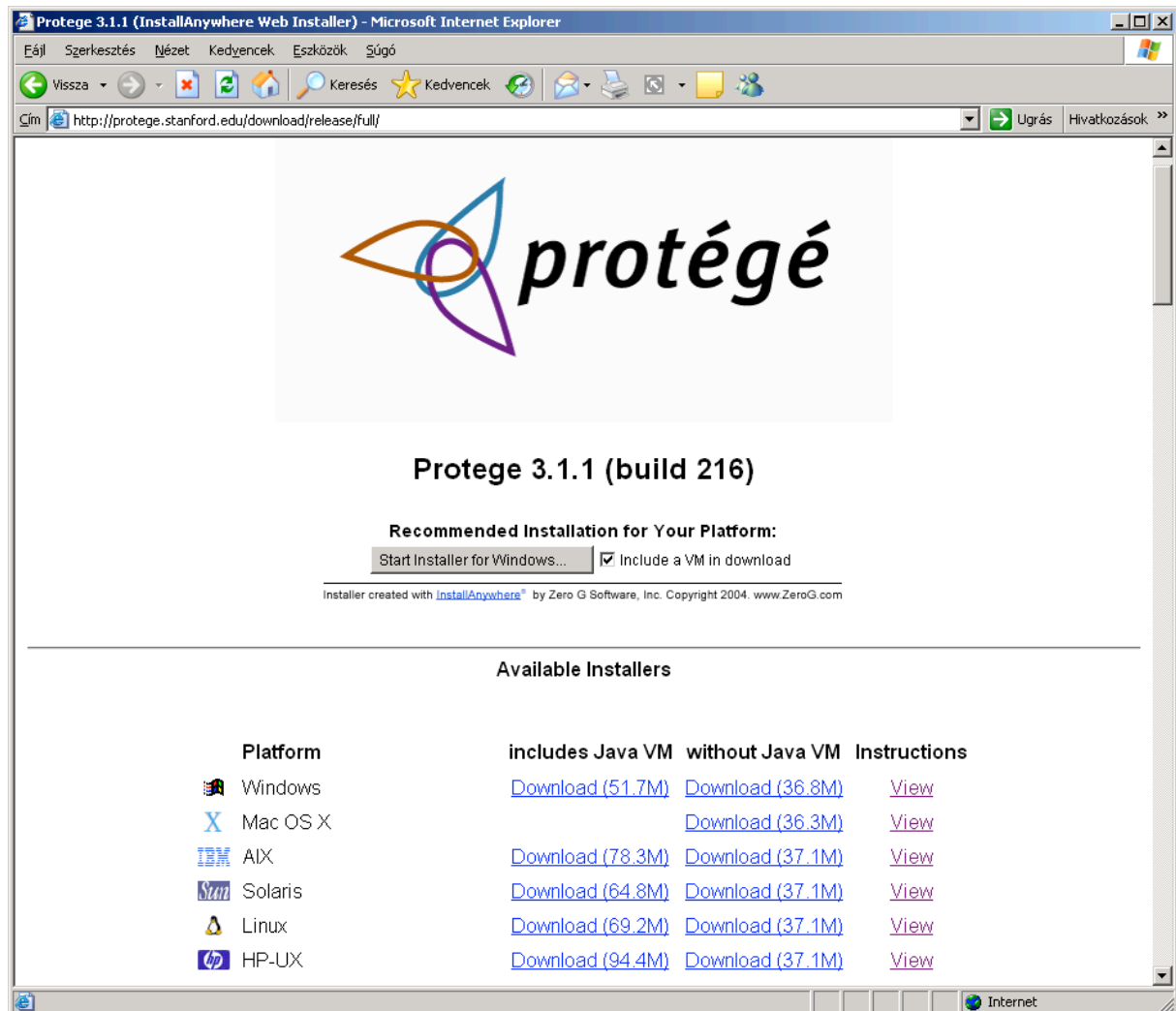


A Protégé programcsomagok és kiegészítők a <http://protege.stanford.edu> URL-ről elindulva tölthetők le. A **Downloads** menü kiválasztása után az üzemeltetők kérik az új



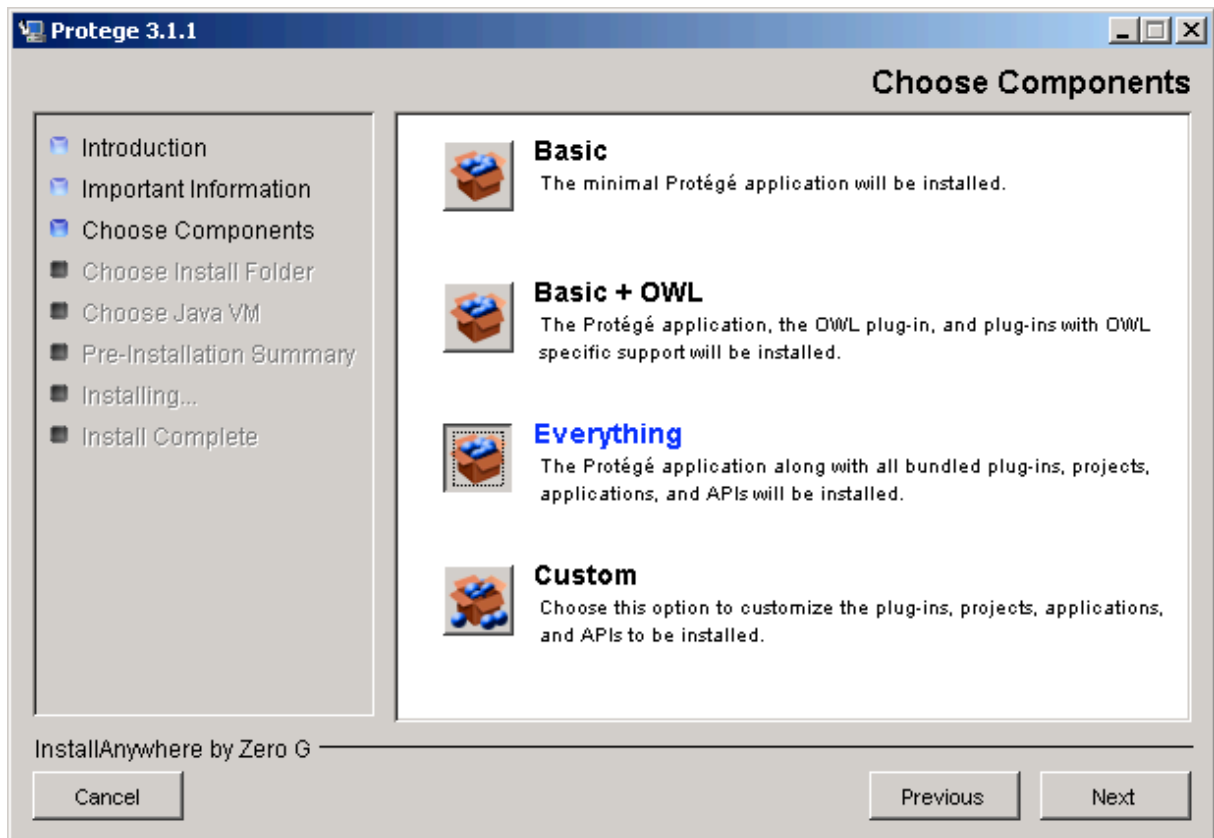
felhasználókat, hogy regisztráljanak (ezt érdemes megtenni, hiszen a rendszer használata ingyenes, a regisztráció arra szolgál, hogy pályázatok beadása esetén lehessen érvelni a felhasználók nagy számával).

A letöltéshez az egységesség és a kompatibilitás miatt célszerű lenne, ha minden egyes felhasználó a (*Download full Protégé release version 3.1.1 (recommended)*) változatot töltené le, és ezt telepítené. Az egyes verziók között megváltozhat a fájl formátum, adatbázis formátum, kliens-szerver protokoll, stb. ami egy bonyolult migrálási problémához vezet.

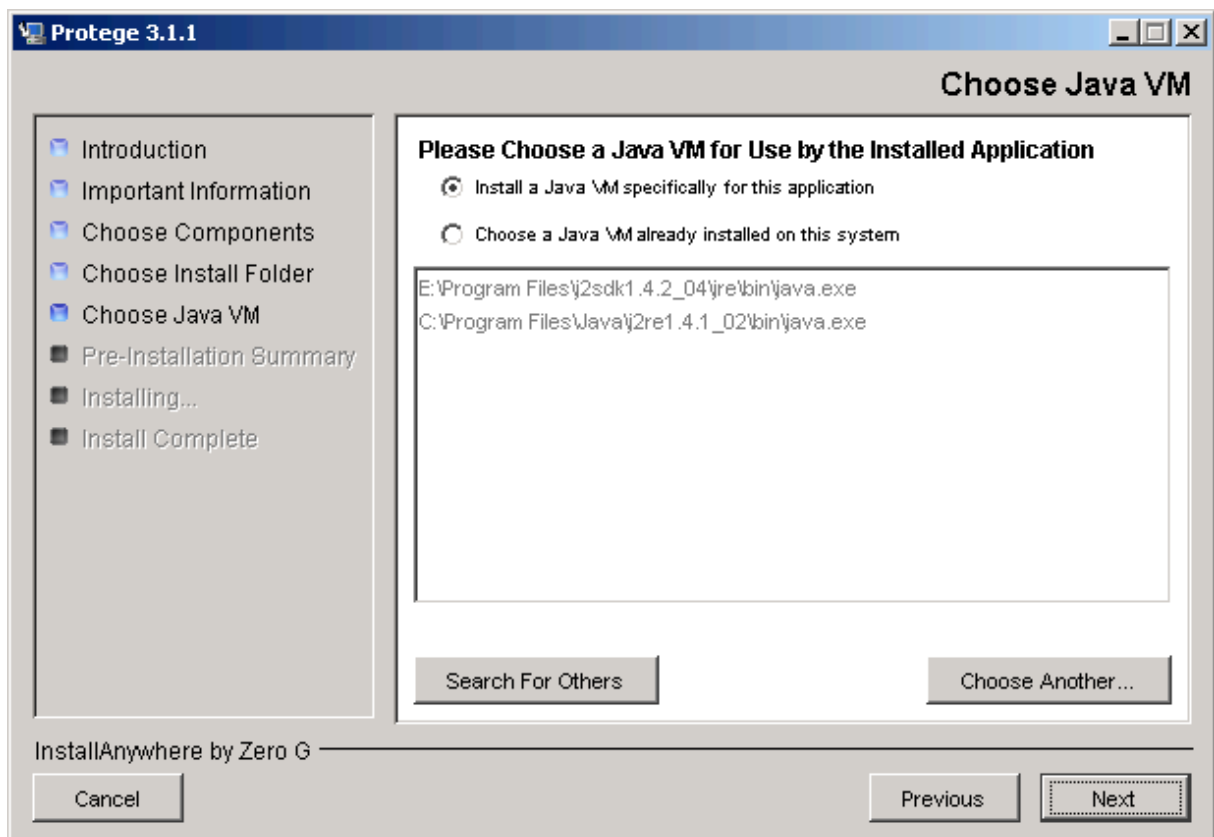


Az alábbi ábrán látható, hogy számtalan platformra készült el a rendszer, köszönhetően az alkalmazott Java technológiának, amely a portolást jelentős mértékben megkönnyíti. Célszerű a telepítőt a Java VM-mel (virtual machine) együtt letölteni, mert egyébként előfordulhat, hogy a számítógépre korábban telepített Java és a Protégé által igényelt Java verziók között probléma merül fel, amit elég nehéz detektálni, és megfelelően kezelni. Telepítésnél is a Protégé által a csomagba betett Java-t célszerű használni, az egyszerűség kedvéért.

A telepítő csomag elindítása után célszerű a telepítést úgy végrehajtani, hogy az összes, a csomagban szereplő anyag kerüljön fel a számítógépre. Így a későbbiekben nem kell a hiányzó bővítményekkel bajlódni. A „Basic” telepítés nem tartalmazza a Protégé szerverét, ezért erre kifejezetten ügyelni kell a szerver telepítésekor. Ilyen esetben az „Everything” vagy az egyedi (Custom) telepítést kell választani. Egyedi telepítés esetén nem alapértelmezett a szerver, ezért ezt akkor kézzel ki kell választani.



Amennyiben a csomagban található Java virtuális gépet kívánjuk használni, úgy válasszuk ki az „*Install a Java VM specifically for this application*” rádiógombot. Ha biztosak vagyunk a gépen található Java megfelelő, akkor meg kell adnunk azt az alkönyvtárat, ahol a rendszer megtalálja.



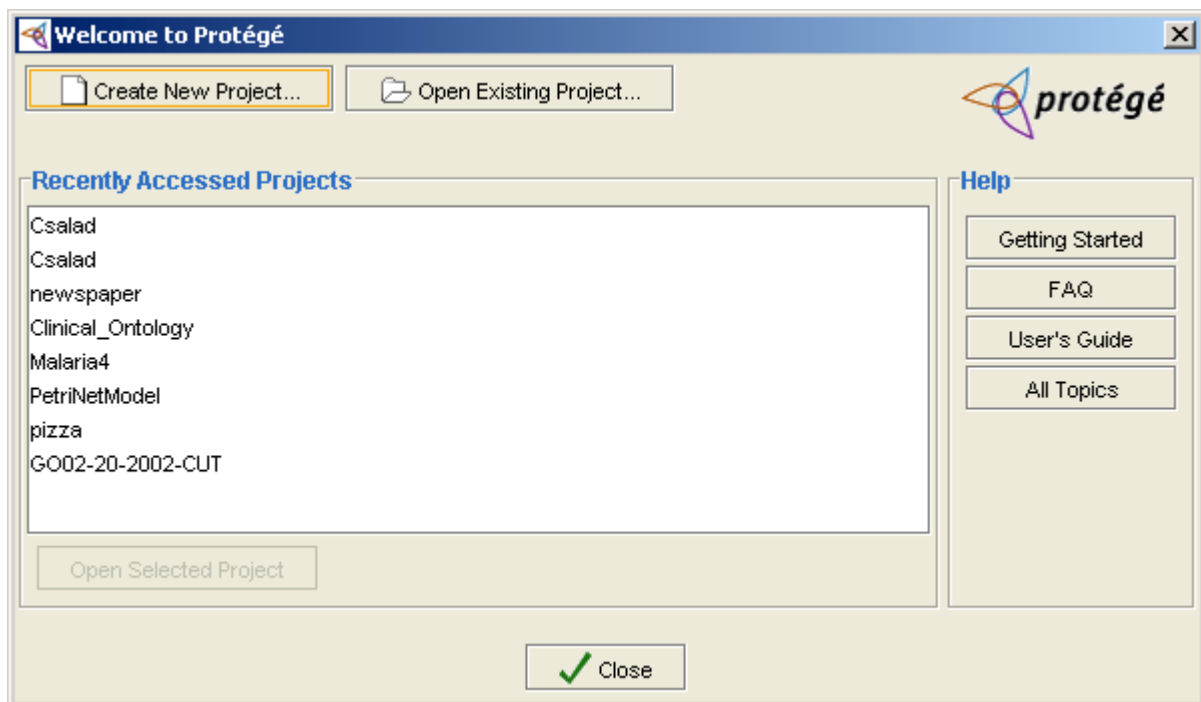
## 2. OWL adatbázis projektek létrehozása

A Protégé nemcsak natív projekteket tud kezelni, hanem éppen a bővítményei révén, OWL projektek szerkesztését is lehetővé teszi. Ennek érdekében tud OWL fájlokból projektet importálni, meglévő Protégé projektet OWL formában exportálni. Egy további bővítménnyel pedig SQL adatbázistáblába is ki tudja menteni egy OWL projekt adatait. Ezt OWL adatbázisnak nevezik. Az adatbázis egyetlen táblát tartalmaz, amelynek sorai írják le az ontológia osztályait, relációit.

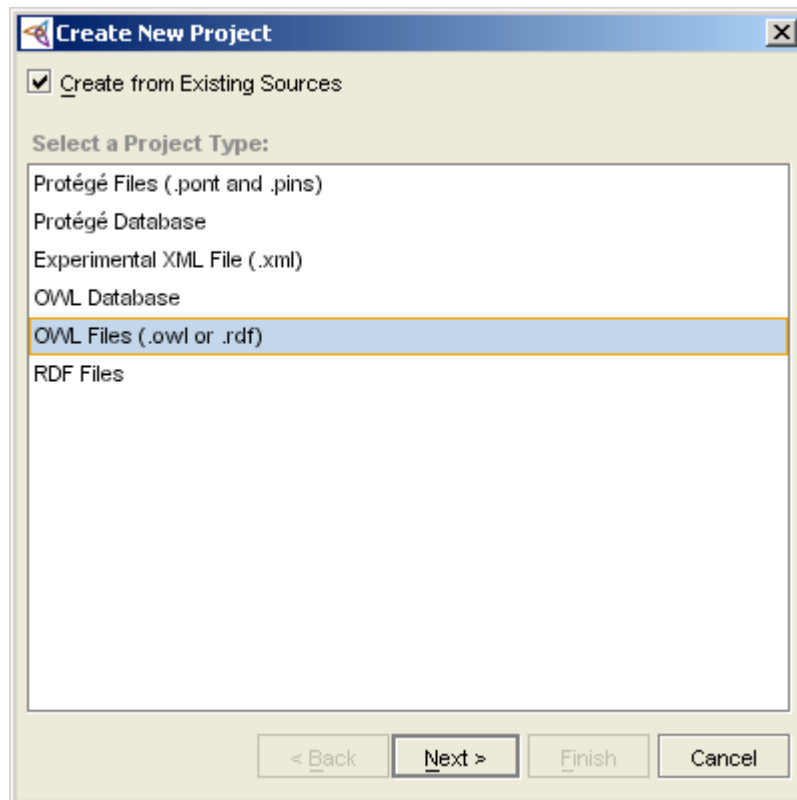
A Protégében a megnyitott projekteket egy .pprj (Protégé PRoject) fájlal reprezentálja. Ez OWL projektek esetén tartalmazza azt, hogy az eredeti OWL reprezentáció honnan származik (adatbázis, vagy OWL fájl), hogyan tud a rendszer az eredeti OWL állományhoz hozzáférni (azért, hogy a végén, vagy szerkesztés közben menthesse az adatokat OWL formában is a rendszer), továbbá az OWL fájl tartalma is benne van a fájlban Protégé formára konvertálva. A későbbiekben nem szükséges mindig OWL fájlból létrehozni a projektet, elegendő a .pprj fájl megnyitni és a fájl tartalmából a rendszer tudni fogja, hogy ez egy OWL projekt, és a kimentett adatbázisból, OWL fájlból beolvassa az adatokat.

### 2.1. OWL projektek importálása

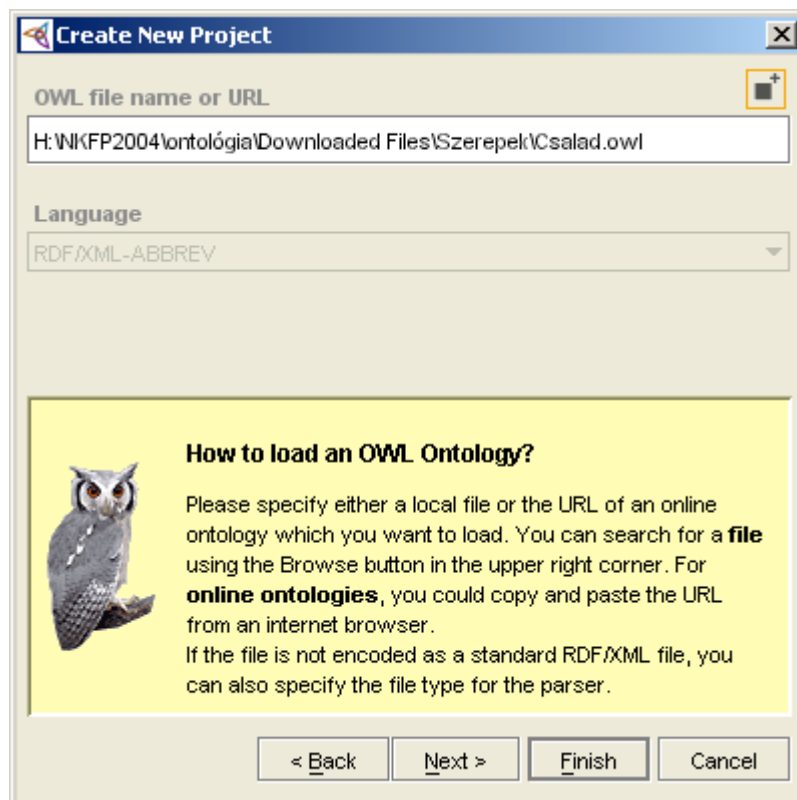
Amennyiben egy OWL fájlba kimentett ontológiát szeretnénk betölteni a Protégébe, akkor a következőképpen kell eljárunk. Egy új Protégé projektet kell létrehoznunk, amely OWL fájlból olvassa be az adatokat. A Protégé elindítása után megjelenő dialógus ablak segítségével lehet már létező projektet megnyitni, illetve új projektet létrehozni.



A „*Create New Project*” gombra kattintva kiválaszthatjuk, hogy a projektet milyen meglévő reprezentációból szeretnénk felépíteni. Lehetőség van régebbi verziójú Protégé fájlokat választani, egy kísérleti XML formát, OWL fájlt, OWL adatbázist, illetve RDF fájlt megadni. Ha már rendelkezünk egy OWL fájlal, akkor ne feledkezzünk el kiválasztani a „*Create from Existing Sources*” checkboxot, ekkor megadhatjuk azt a fájlt, amelyből az importálás történik.

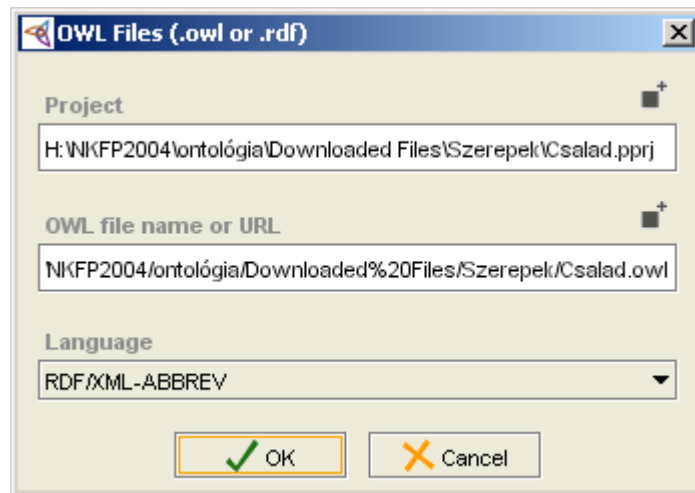


A „Next” gombra kattintva választhatjuk ki az OWL fájlt.



## 2.2. OWL projektek mentése Protégé formában

Amikor az importálás megtörtént, akkor a rendszer automatikusan létrehoz egy az OWL fájl nevével megegyező, de .pprj kiterjesztésű Protégé projekt fájlt is. Ha korábban volt már ilyen fájl, akkor azt kérdés nélkül felülírja. Erre érdemes gondolni.



A munka befejeztével, vagy menet közben a projektünket a „Save As” menü kiválasztásával menthetjük ki más néven. Ha megfelel az a mód, ahogy a rendszer alapértelmezetten működik, tehát, hogy az OWL fájl nevével egyező .pprj projekt fájlt hoz létre, akkor egyszerűen csak a „Save” menüt válasszuk ki.

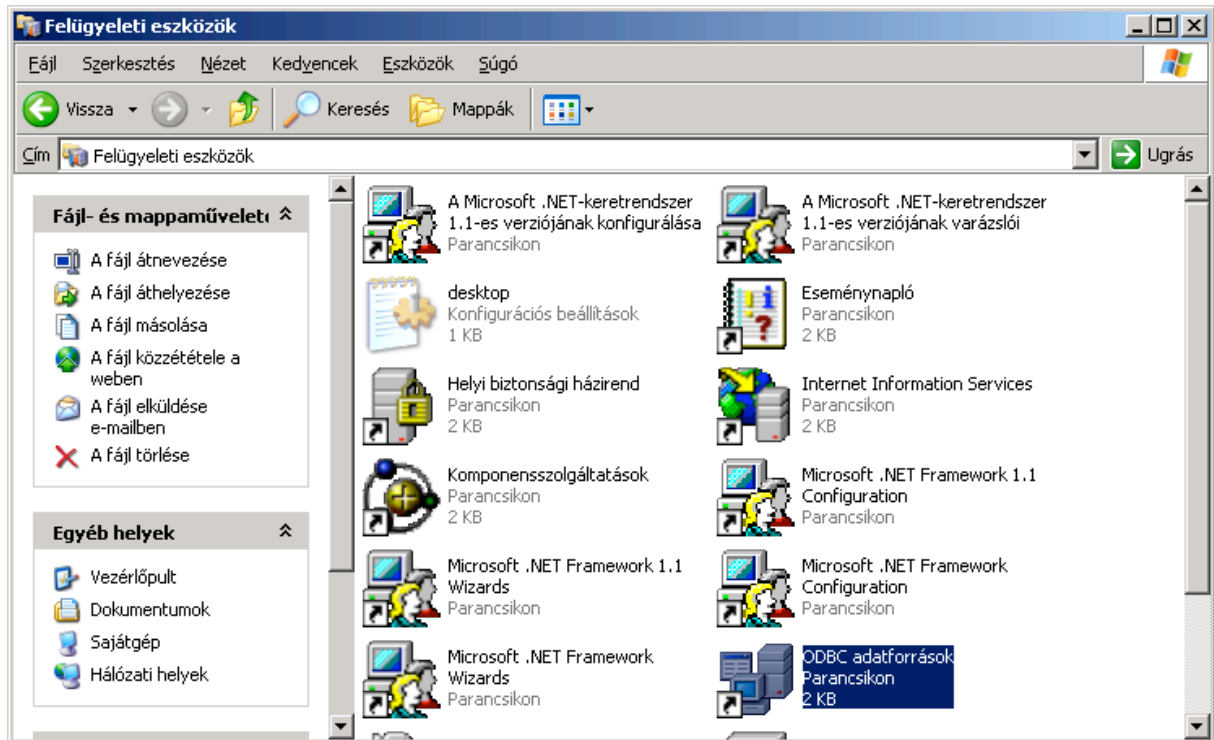
### 2.3. OWL projektek mentése OWL adatbázisba

Előfordulhat, hogy az eredetileg OWL fájl formában lévő projektünket SQL adatbázisban kívánjuk a továbbiakban tárolni. Ez annál is inkább fontos, mivel a Protégé szerver jelen állapotában nem tud OWL fájl alapú projekteket kezelni, csak OWL adatbázisból tud adatokat importálni. Az adatbázis kezelés Protégében többféle módon megvalósulhat: eredeti Java JDBC driverekkel, ODBC driverek segítségével, amelyeket egy JDBC-ODBC illesztőfelület kapcsol össze, végül saját Javában írt modulokkal. Az első két változat a Protégé felhasználók körében elterjedten használt és működő megoldás, addig az utolsó nem igazán célszerű. Általában az ingyenes adatbázisrendszerek használata esetén szerezhető be natív Java JDBC szoftverek, pld. mySQL, postgresSQL; a kereskedelmi célú rendszereknél ezt meg kell vásárolni. A JDBC-t használó megoldások is széles körben használtak, azonban most a JDBC-ODBC híd felhasználási lehetőségeit ismertetjük.

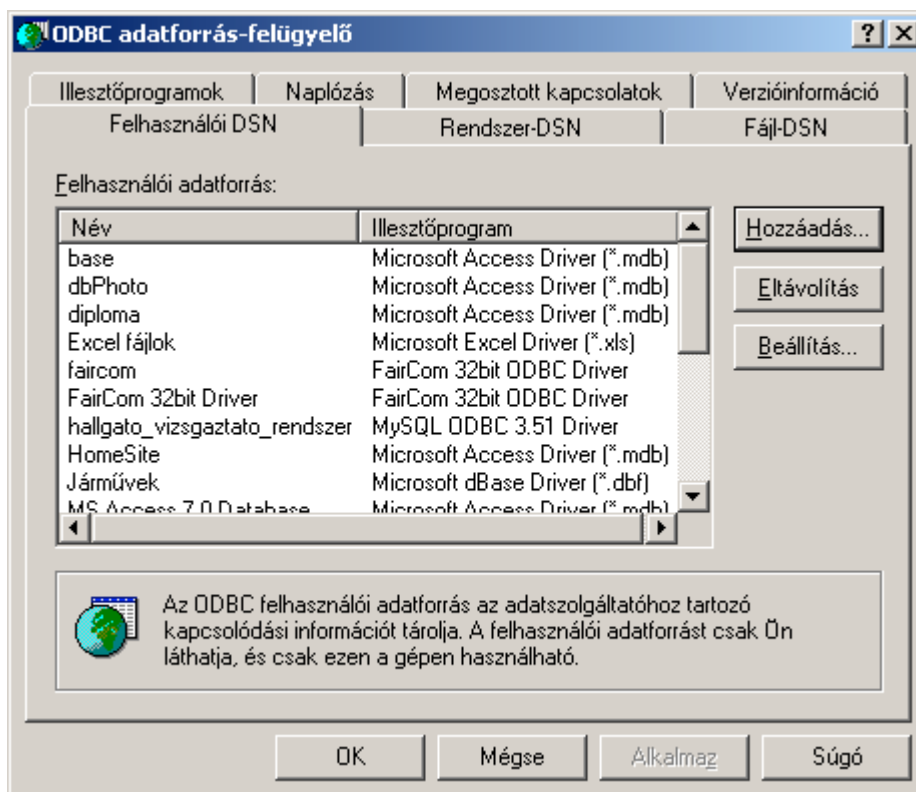
### 2.4. MS Access-szel megvalósított OWL adatbázis

Az Microsoft DataBase technológia, amelyet az MS Access felhasználói programmal tettek könnyen használhatóvá már régóta széles körben ismert. A használata egyszerű, az adatbázist egyetlen sima fájl tárolja (flat file database system). Az adatbázishoz különböző jogokkal rendelkező felhasználókat lehet rendelni, több táblát kezel, egyszerre több lekérdezés futtat rajta, mivel a meghajtó DLL-ek támogatják a többszörös folyamatokat. A kezdeti, üres adatbázist az ODBC adminisztrátor programmal kell létrehozni, amely a vezérlőpulton keresztül, a *Felügyeleti eszközök* mappában található.

Az ODBC használat során **adatforrásokat** kell létrehozni, amely az adatbázis egy virtualizált formája. Az ODBC adminisztrátorral lehet adatforrást létrehozni és egy konkrét megvalósítással, jelen esetben egy MS Access .mdb fájjal összekapcsolni. Az adatforrásokat a nevük (egy C string) azonosítja. A technológia a konkrét adatbázis megvalósítást lényegében elrejt a felhasználók elől, a programok egy virtuális adatbázison dolgoznak már.

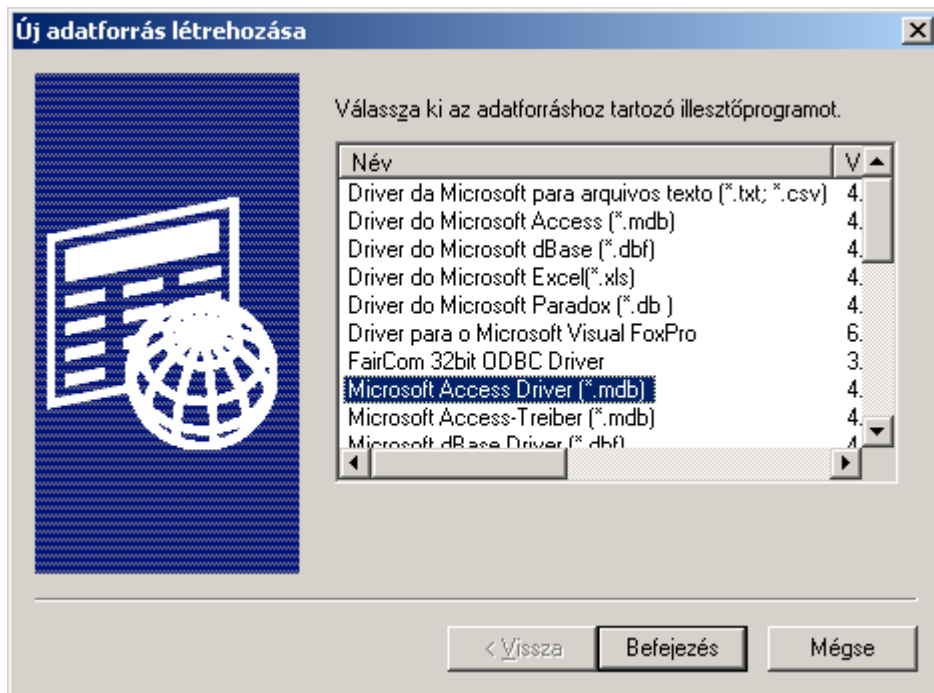


Új adatforrás létrehozását a „Hozzáadás” nyomógomb lenyomásával kezdeményezzük.

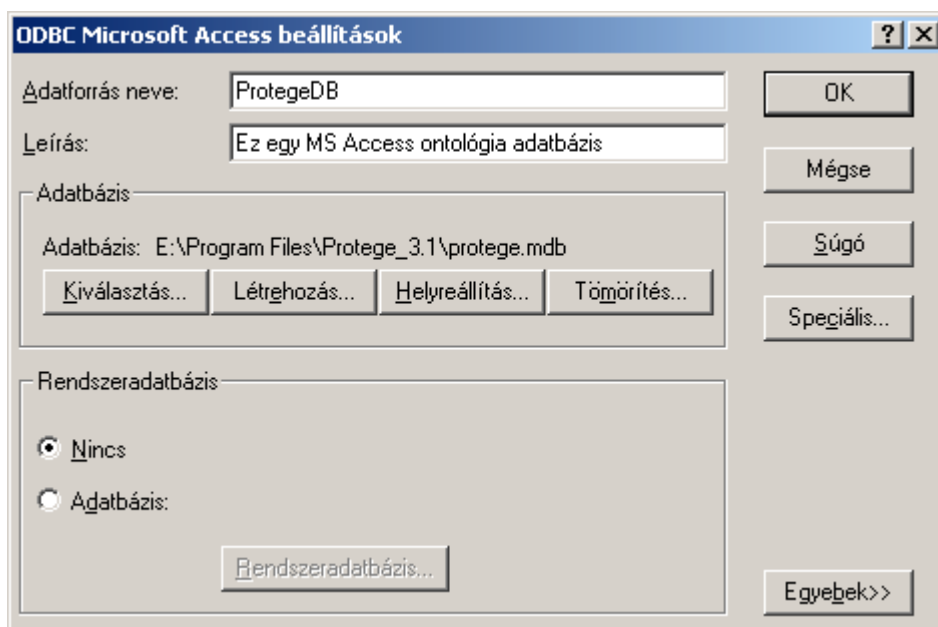


A hozzáadás első lépése az adatbázis megvalósítás kiválasztása. Jelen esetben a „Microsoft Access Driver (\*.mdb)”-t válasszuk ki.





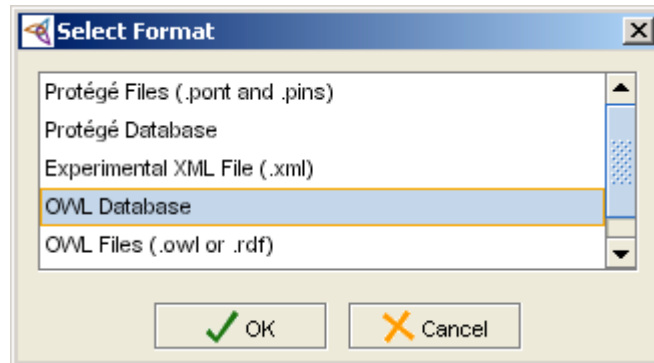
A következő lépés az adatforrás nevének megadása (ProtegeDB), egy magyarázó megjegyzést is adhatunk a definícióhoz. A „Kiválasztás” nyomógommbal megadhatjuk az adatbázist tároló fájlt, az alkönyvtárat és a fájl nevét. Létező adatállományt is választhatunk, akkor a későbbiekben ezt fogjuk kibővíteni, módosítani. A Protégé az adatbázisban egyetlen adattáblát hoz létre. Mentéskor ezt előbb kitörli, majd teljesen új rekordokkal tölti fel, a memóriában tárolt ontológia alapján. A tábla nevét megadhatjuk majd, így egy .mdb fájl több kimentett ontológiát is tartalmazhat. Az MS Access adatbázis több felhasználót is megenged, ezért egy adatbázisban felhasználónként is külön-külön ontológia változatokat tárolhatunk.



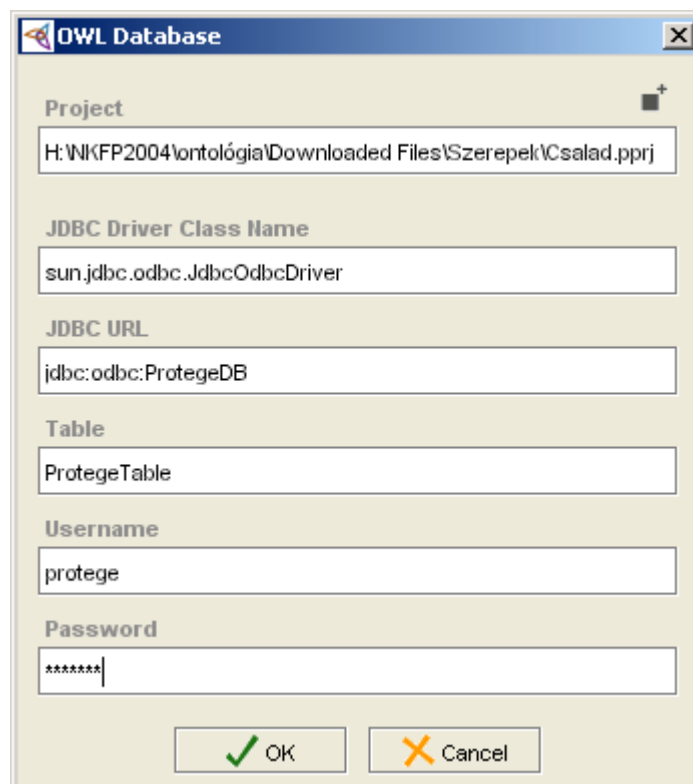
Ha eddig még nem rendelkezünk adatbázis fájlal, akkor a „Létrehozás” gombbal lehet egy üres 64kB méretű kis adatbázis fájlt létrehozni. Ebben az esetben nincsenek megkülönböztetett felhasználók. Több felhasználó esetén a hozzáférési jogok és jelszavak adminisztrálására egy ún. rendszeradatbázist is létre kell hozni (egy mindenki által

használható adatbázisnál erre nincs szükség). Az itt kiválasztott vagy létrehozott állományokat lehet azután a későbbi feldolgozások során bővíteni.

A Protégéből a most létrehozott adatforrásba a következőképpen lehet a projektünket kimenteni: A „*File / Convert Project to Format ...*” menüt válasszuk ki. A megjelenő listában pedig válasszuk az OWL Database formát.



A következő képernyőn az adatbázishoz történő kapcsolódás paramétereit kell beállítanunk. Mivel ez a menüpont a projekt formátumát konvertálja át, ezért a továbbiakban a projekt .pprj állománya már azt fogja tartalmazni, hogy ez a projekt egy OWL adatbázis projekt. Amennyiben ezt nem szeretnénk, úgy az első, a Project sorban megadott útvonalat és nevet változtassuk meg, ekkor a Protégé egy teljesen új nevű, OWL adatbázis projektet hoz létre, a régit pedig meghagyja eredeti formájában.

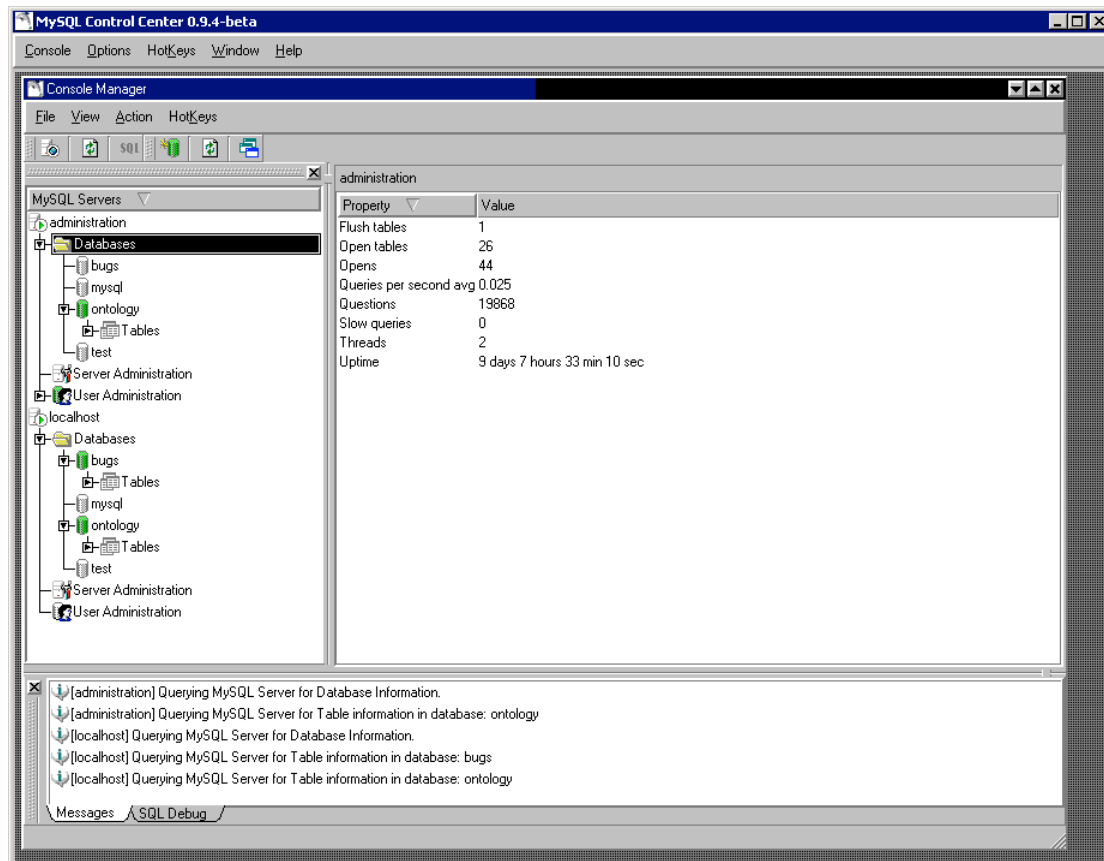


Ezután a JDBC Driver Class Name sorba a következőt kell beírni: **sun.jdbc.odbc.JdbcOdbcDriver**. A JDBC URL ilyenkor: **jdbc:odbc:ProtegeDB**, ahol az ODBC adatforrás neve a ProtegeDB. Az adatbázistábla neve a következő – ez szabadon választható –, majd a felhasználói azonosító és jelszó, amivel kapcsolódhatunk a rendszerhez. Ha nem hoztunk létre Rendszeradatbázist, akkor az MS Access átenged minden kérést, nem ellenőriz semmilyen Username-t és Password-t, így ide mindegy, hogy mit írunk, a rendszer

nem fogja figyelembe venni. Ezzel létrehoztunk egy OWL adatbázis projektet, amelyet a következőkben már az első sorban megadott projekt fájl betöltésével tölthetünk be.

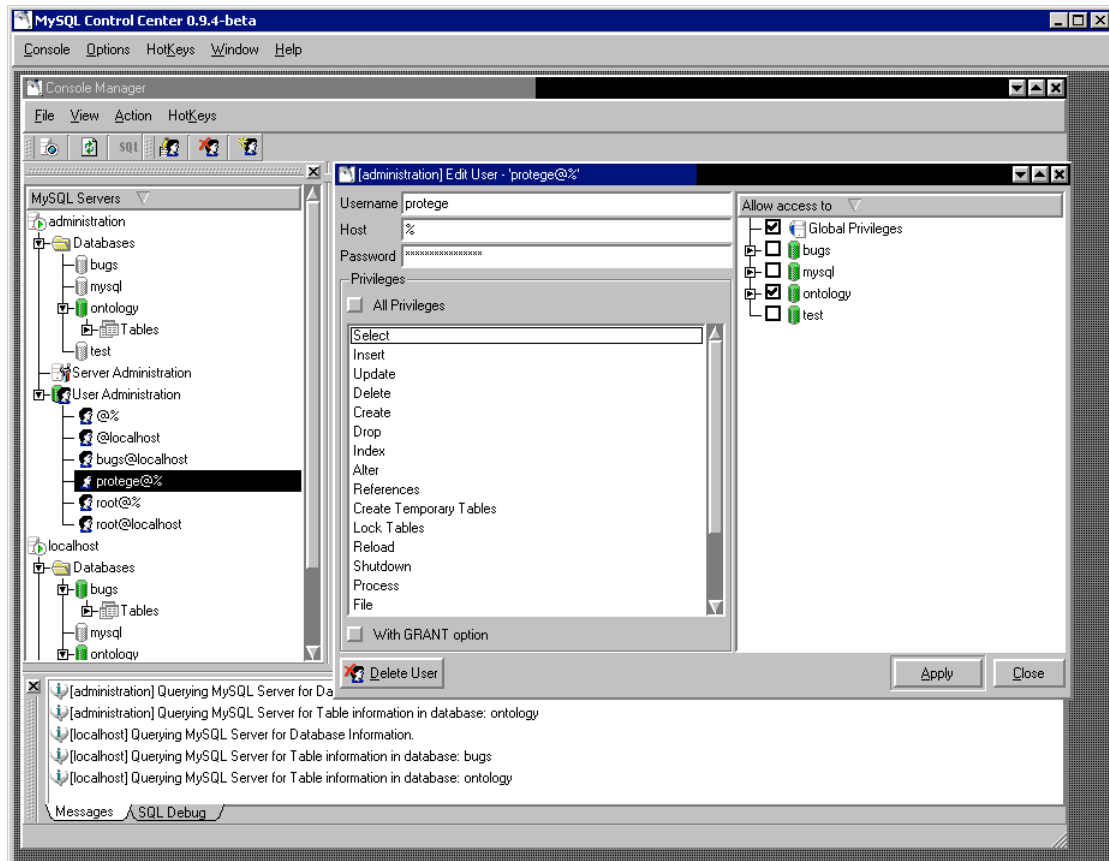
## 2.5. mySQL-lel megvalósított OWL adatbázis

A teendők nagyjából hasonlóak, azzal a különbséggel, hogy itt bizonyos szerver oldali feladatok is lehetnek: adatbázist és felhasználót kell létrehozni, a felhasználó jogait be kell állítani.



A MySQL Control Center nevű programmal lehet a legkényelmesebben ezeket a feladatokat elvégezni. A Databases mappára a jobb egérgombbal kattintva lehet új adatbázist létrehozni, amennyiben szükséges. A „User Administration” mappára kettőt kattintva a felhasználók és jogaik beállítására szolgáló alrendszerhez juthatunk el. A mySQL az adatbázisok egy egyszerűbb megvalósítása esetén, a rendszer az egyes adatbázisokat külön fájlokban tárolja (amelyek igény szerint megnyúlnak). A fájlokat egy előre adott alkönyvtárban helyezi el, amit telepítéskor kell megadni.

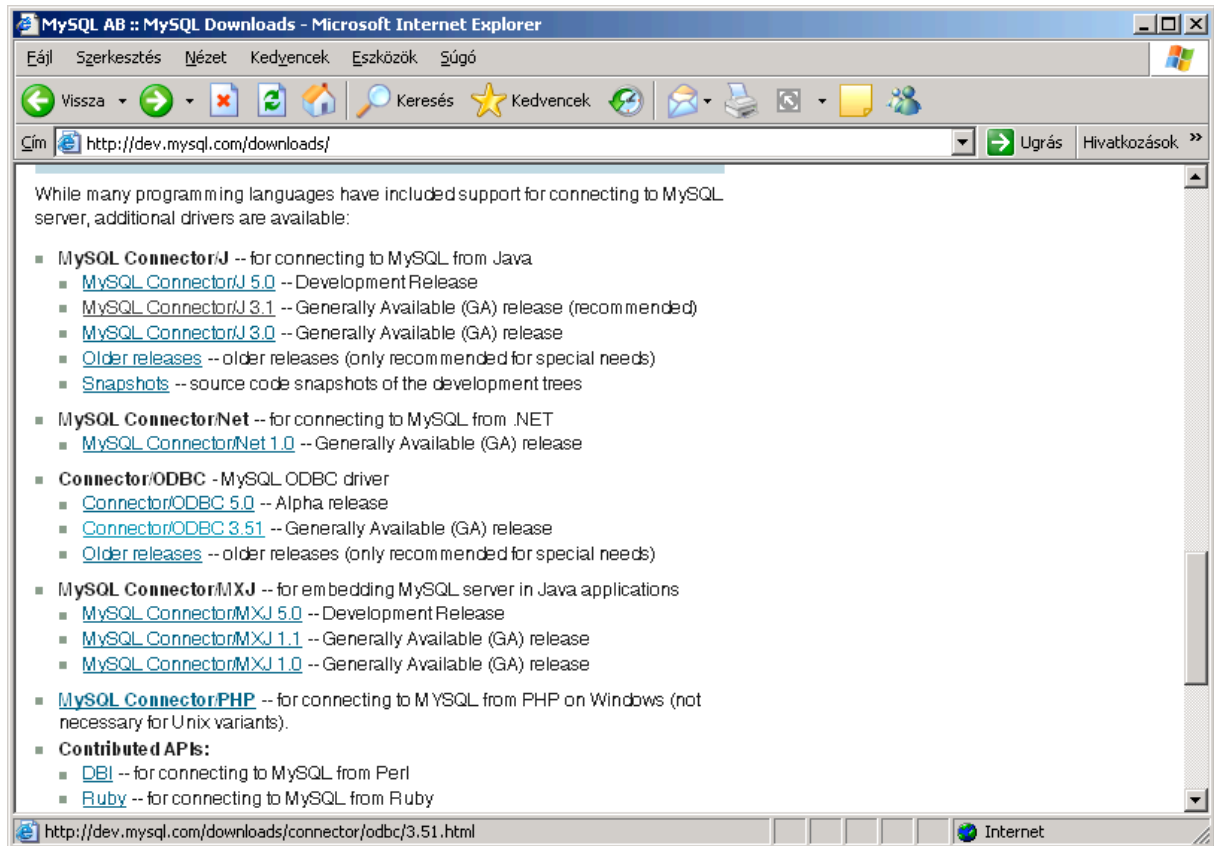
A mySQL felhasználók esetében egyedileg lehet szabályozni, hogy ki melyik adatbázishoz férhet hozzá, és milyen lekérdezési, módosítási és egyéb privilégiumai vannak. Az ontológiákat célszerű egy külön fájlban (adatbázisban) szeparáltan tárolni, és célszerű egy külön erre a célra létrehozott Usert rendelni hozzá. Az adatbázistábla nevét a Protégében megadhatjuk, ezért ebben az adatbázisban ezzel a felhasználóval is több ontológiát tudunk tárolni és használni.



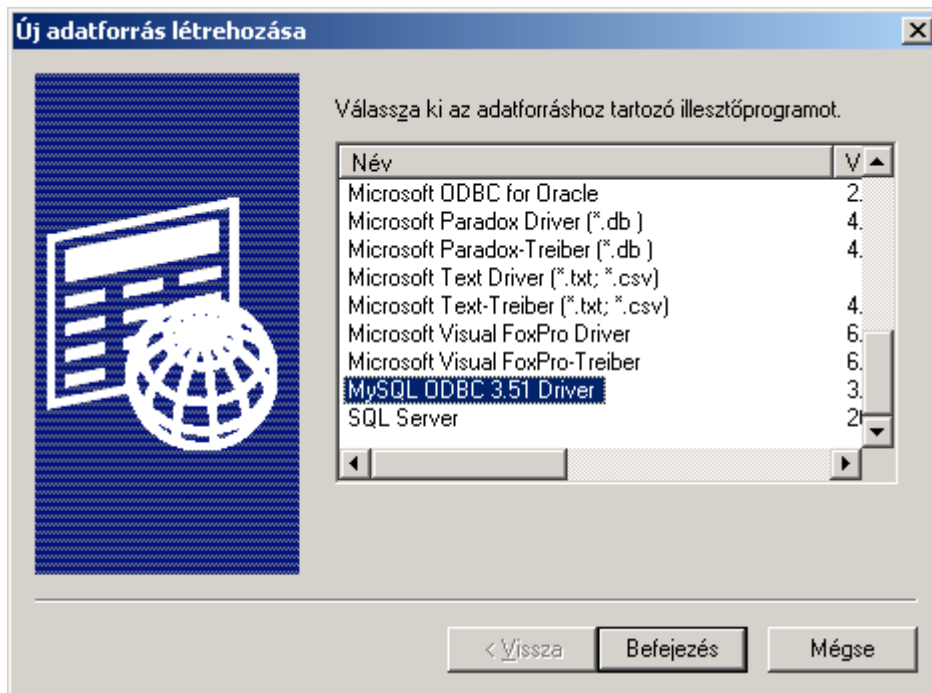
A mySQL használatához ODBC felületen keresztül szükség van a mySQL ODBC driverre, amely a <http://www.mysql.org> URL-ről letölthető.



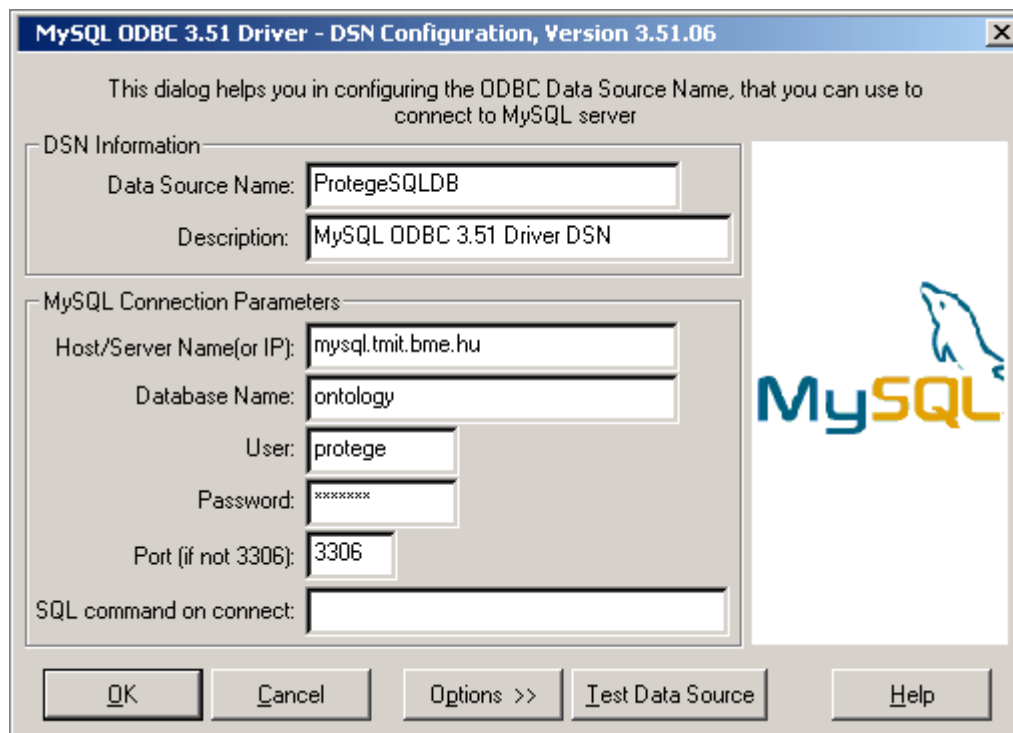
A Downloads menüt kell kiválasztani, majd pedig a Connector/ODBC felsorolásból az Connector/ODBC 3.51 – verziószámú meghajtó csomagot. A további képernyőkön kiválaszthatjuk a platformot (Windows, Linux, HP-UX, Solaris stb.). Letöltés után ezt az anyagot a kívánt gépre telepíteni kell.



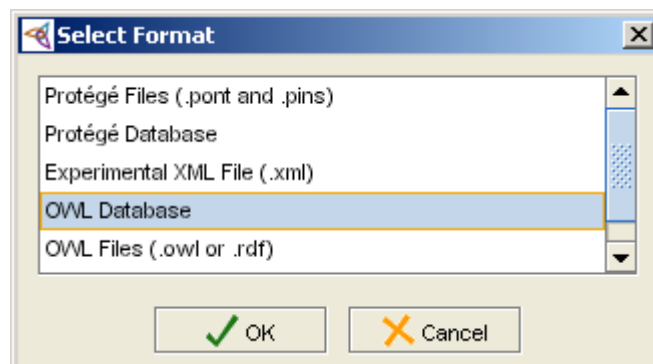
Az ODBC adminisztrátor programban ebben az esetben a MySQL ODBC 3.51 Drivert kell kiválasztani.



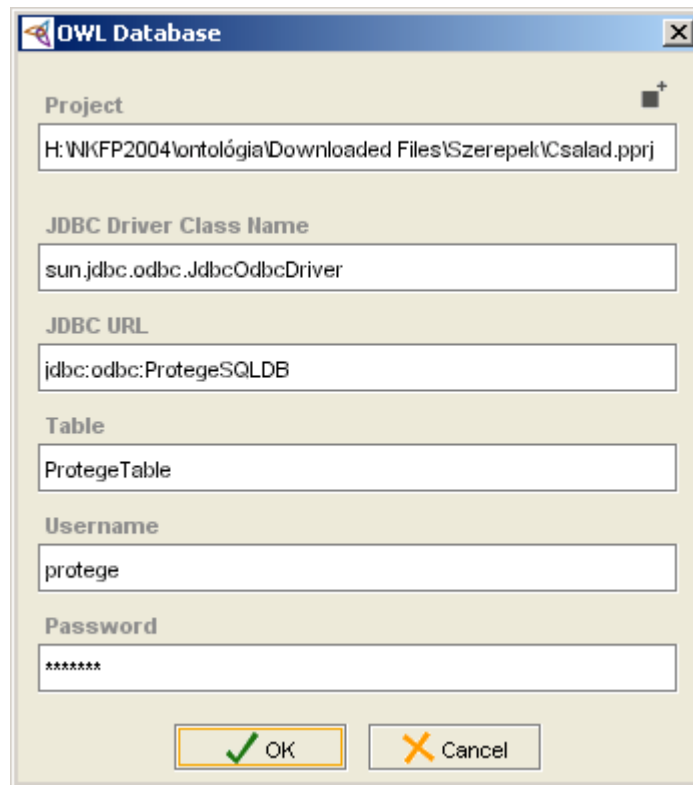
Az adatforrás adatainak beállítása ebben az esetben már bonyolultabb. Itt is kell nevet adni az adatforrásnak, illetve egy szöveges magyarázatot hozzá. Ezekon kívül a szerver hostnevét, az adatbázis nevét, a belépési useridet és a jelszót, a TCP/IP portot kell megneveznünk. A kapcsolat működőképességét a „*Test Data Source*” gombbal célszerű ellenőriznünk.



A Protégéből ebbe az adatforrásba menteni a projektünket ugyancsak a „*File / Convert Project to Format ...*” menüvel lehet. A megjelenő dialógus ablakból válasszuk ki ugyancsak az OWL adatbázis formátumot.

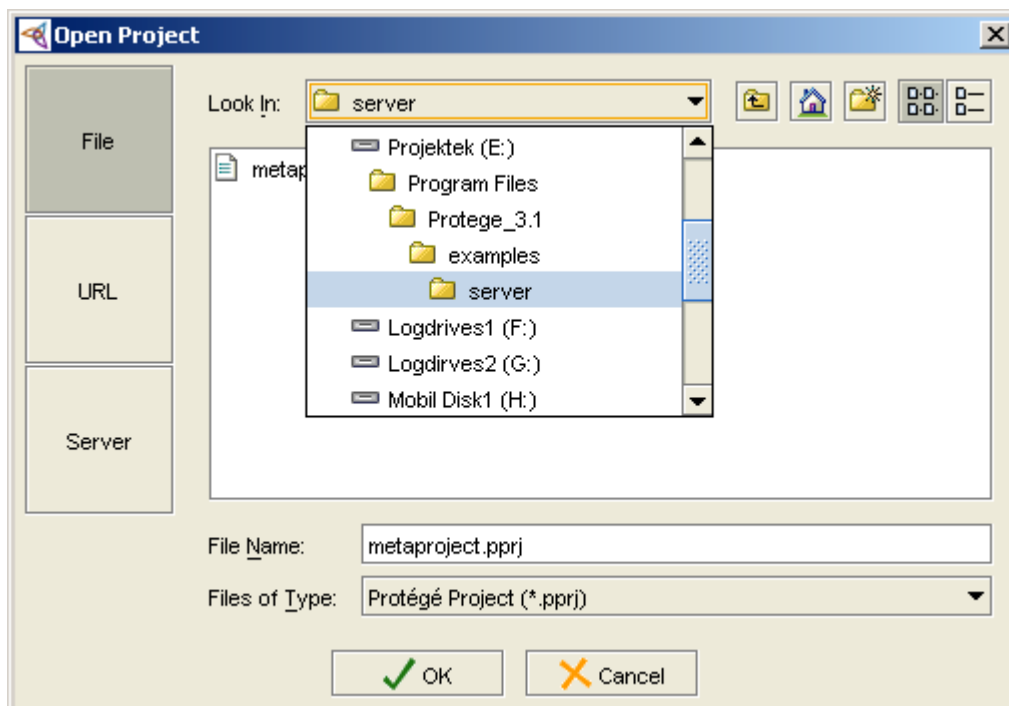


A továbbiakban pedig hasonlóan a korábban már ismertetett eljárásához, adjuk meg az adatbázishoz kapcsolódás paramétereit. Az első, a Project sorba az átkonvertált projekt fájl nevét írjuk be. A JDBC Driver Class Name ugyanaz: **sun.jdbc.odbc.JdbcOdbcDriver** lesz. Az útvonal (URL) pedig: **jdbc:odbc:ProtegeSQLDB**, ahol a ProtegeSQLDB a korábban létrehozott ODBC adatforrás neve, amely a mySQL adatbázishoz csatlakozik. Az adatbázistábla neve egy általunk választott táblanév. A Username és a Password helyes kitöltése most fontos, mert a mySQL helytelen paraméterek esetén nem ad hozzáférést az adatbázishoz.



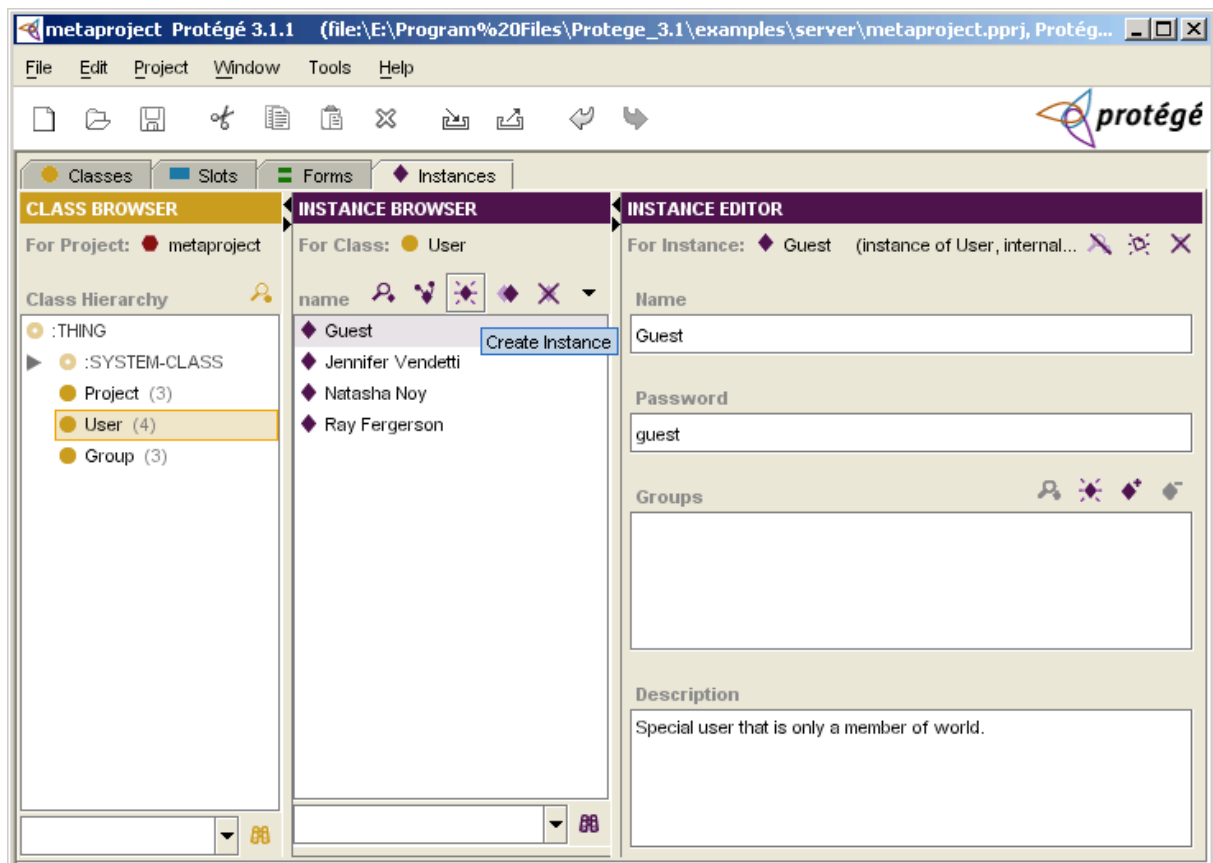
### 3. OWL adatbázis projektek beillesztése a Protégé szerverbe

A Protégé szerver alapvetően natív Protégé projektek szerkesztését támogatja, de korlátozásokkal OWL projektek esetében is használható. OWL fájlkból sajnos nem tud importálni adatokat, azonban arra van lehetőség, hogy OWL adatbázisból töltsön be projekteket. Ezért ez a lehetőséget mutatjuk most be.



A Protégé szerver működésének fontosabb paraméterei egy saját magával készített ontológia projektben (metaproject.pprj) vannak leírva. A projekt a Protégé telepítési könyvtárban található, azon belül is a \Program Files\Protege\_3.1\examples\server könyvtárban.

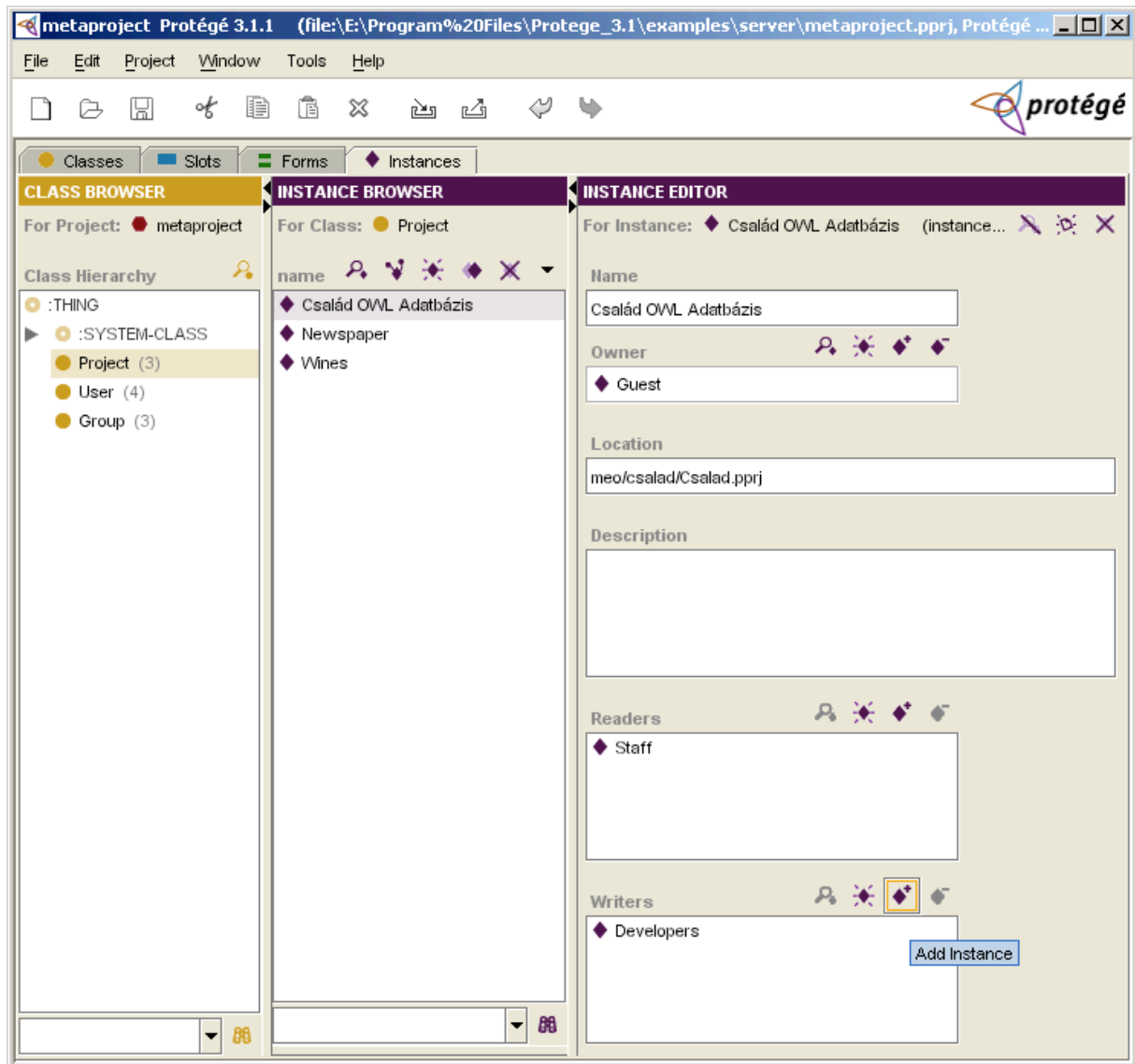
A metaproject fájl három osztályt, a Project, a User és a Group osztályokat tartalmazza és ezek egyes példányait. A szerver által kezelt adatok az egyes példányok adatai. Új szerver projekt létesítésekor egy Project példányt (instance) kell létrehozni, és meg kell adni azt, hogy mely felhasználók kezelhetik az objektumokat. A felhasználók csoportokba vannak beosztva, vannak fejlesztők (Developers), alkalmazottak (Staff) és az összes többi egyéb felhasználó (World). Új felhasználót az alábbi képernyőn lehet felvenni, a „Create Instance” gombra kattintva.



A következőkben megadhatjuk, a felhasználónak mi legyen a neve, mi legyen a bejelentkezési jelszava, hogy melyik csoportokba tartozzon. A jelszót olvasható szöveges alakban kell beírni. Egyelőre a rendszer nem támogatja azt, hogy a begépett jelszót ne lehessen elolvasni, így ez a bizalmasságot veszélyeztetheti.

A Project példány esetében ugyancsak az Instance Browser ablakban a „Create Instance” gombot kell megnyomni, ami után megadhatjuk a projekt adatait. A projekt neve fog megjelenni a kliensek képernyőjén, megadhatjuk a tulajdonos felhasználó nevét, a projekt .pprj fájljának helyét. Jelen esetben ez egy olyan projekt kell legyen, amelyet OWL adatbázis formájúvá konvertáltunk át a korábbi fejezetekben leírt módon. Az útvonalat úgy adjuk meg, hogy a kezdő könyvtár a \Program Files\Protege\_3.1\ könyvtár. Ami ez után (alatt) van azt kell beírni. A csoportok megadásakor már létező csoportokat szűrhatunk be az „Add Instance” gombbal.





Ha az új projekt adatait begépeztük, akkor mentjük el a metaproject.pprj fájlt, és indítjuk el a szerver programot a háttérben. A Protégé szervert a `run_protege_server.bat` eljárás indítja el. Elindítás előtt érdemes megvizsgálni a batch fájl tartalmát, és ha kell módosítani a mentési időket (hány percenként mentsen a rendszer), a lefoglalt memória nagyságát, esetleg útvonalakat a Java virtuális gépéhez. A sikeres elindulást után a szerver betölti a metaproject.pprj fájl alapján a támogatott projekteket és a console ablakban láthatjuk, ha valamelyik projekt betöltésekor hiba lépne fel. Jelen esetben az `INFO: Loading project Család` jelzi, hogy ezt a projektet a rendszer betöltötte, és mivel hibaüzenet nem jelent meg, ezért a továbbiakban használható.

### 3.1. A Protégé szerver által használt TCP/IP portok:

A tapasztalatok szerint a Protégé szerver a következő portokat használja: 1077, 1099 valamint az 1461 számú.

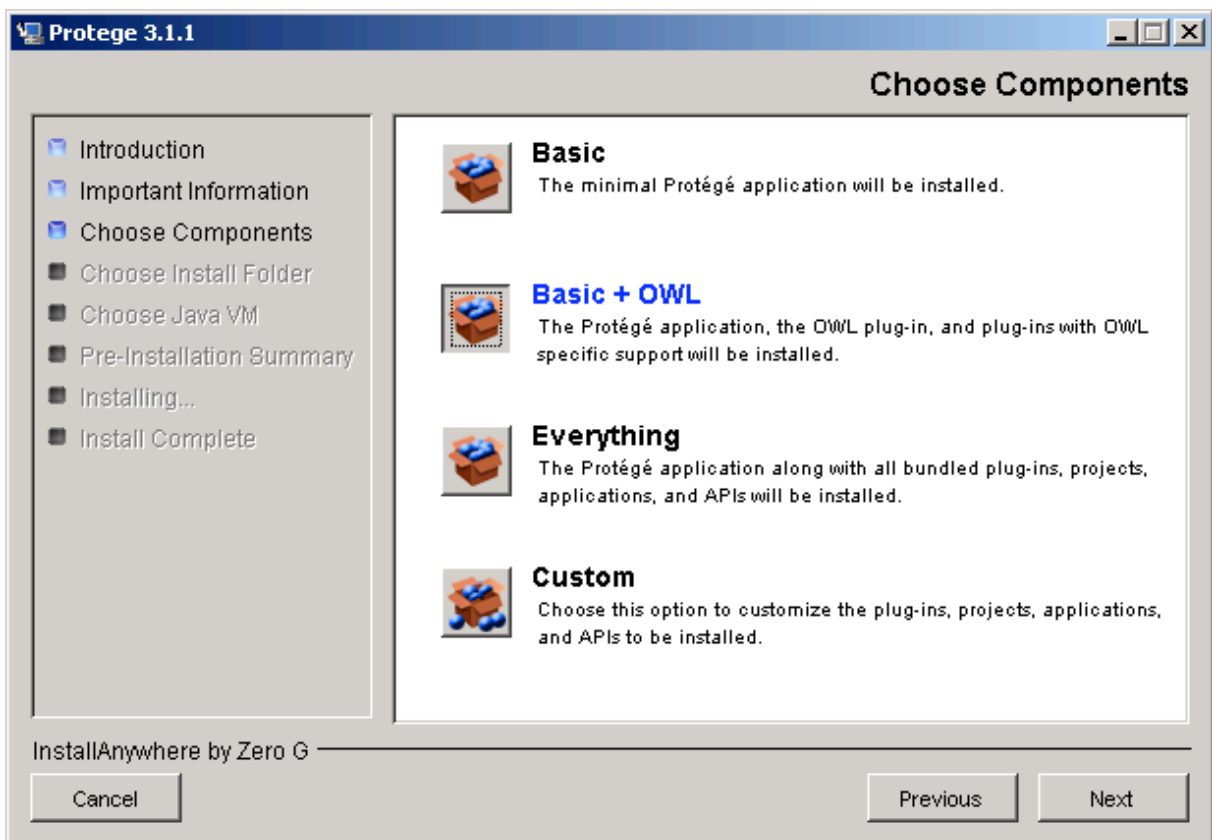
```

Kijelölés C:\WINNT\system32\cmd.exe
CONFIG: Loaded plugin edu.stanford.smi.protege.pal_tabs - PAL Tabs
CONFIG: Loaded plugin edu.stanford.smi.protege.ezpal - EZPal Tab
CONFIG: Loaded plugin edu.stanford.smi.protege.facets_tab
CONFIG: Loaded plugin edu.stanford.smi.protege.instance_tree
CONFIG: Loaded plugin edu.stanford.smi.protege.prompt - PROMPT tab
CONFIG: Loaded plugin edu.stanford.smi.protege.psm - PSM Librarian
CONFIG: Loaded plugin edu.stanford.smi.protege.rdf_backend - RDF Backend
CONFIG: Loaded plugin edu.stanford.smi.protege.uml_tab - UMLS Tab
CONFIG: Loaded plugin edu.stanford.smi.protege.wordnet - WordNet Tab
CONFIG: Loaded plugin edu.stanford.smi.protege.xml_tab - XML Tab
CONFIG: Loaded plugin edu.washington.datagenie
CONFIG: Loaded plugin JessTab - JessTab Plugin
CONFIG: Loaded plugin org.algernon - Algernon
CONFIG: Loaded plugin protege.prologtab
CONFIG: Loaded plugin uk.ac.ecs.iam.akt.tgviztab - TGVizTab
CONFIG: Loaded plugin uk.ac.man.cs.mig.coode.owldoc
CONFIG: Loaded plugin uk.ac.man.cs.mig.coode.owlviz
CONFIG: Loaded plugin uk.ac.man.cs.mig.coode.protege.wizard - Basic Wizards
CONFIG: Loaded plugin uk.ac.man.cs.mig.coode.protege.wizard.owl - OWL Wizards
CONFIG: Installed ProtegeRmiClassLoaderSpi
INFO: Loading project Csalbd
INFO: Loading project Newspaper
INFO: Loading project Wines
INFO: Protege server ready to accept connections...

```

#### 4. A Protégé 3.1.1 kliensek telepítése

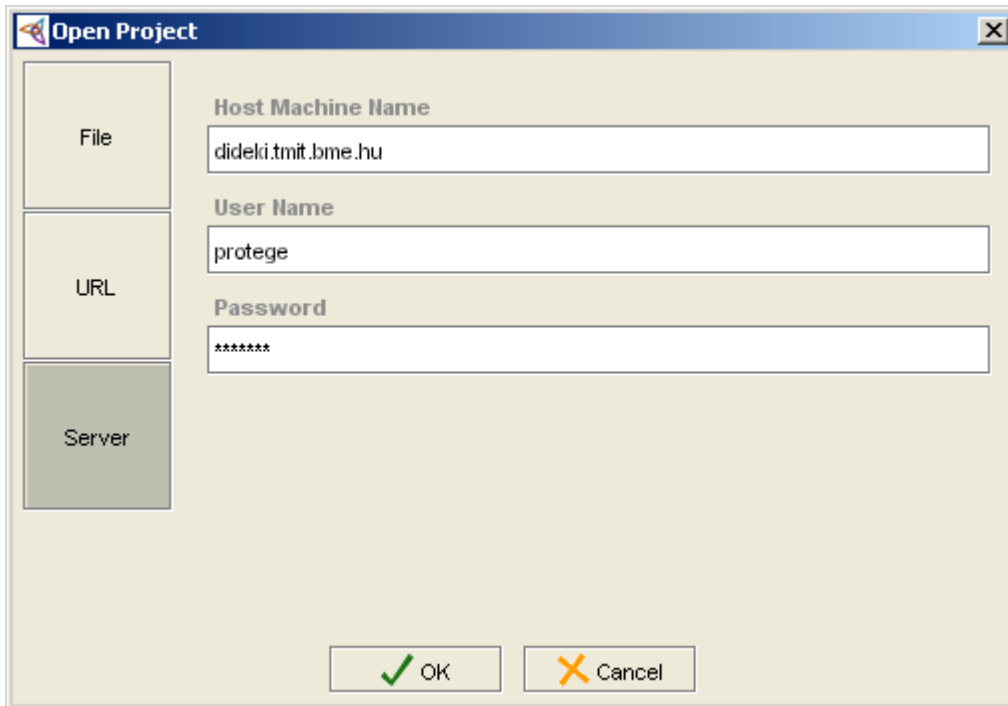
A Protégé kliensek telepítéséhez ugyancsak a *(Download full Protégé release version 3.1.1 (recommended))* változatot kell letölteni. Azonban nem szükséges minden funkciót telepíteni, azaz elegendő lehet a *Basic+OWL*, vagy a *Custom* (egyedi) változat. Az egyedi telepítésnél nem szükséges a szerver telepítése, továbbá szabadon dönthetünk arról, hogy mely bővítményt kívánjuk telepíteni.



Ami a Java virtuális gépet illeti, ajánlott a csomagban meglévő Java verzió használata. Ettől csak indokolt esetben célszerű eltérni.

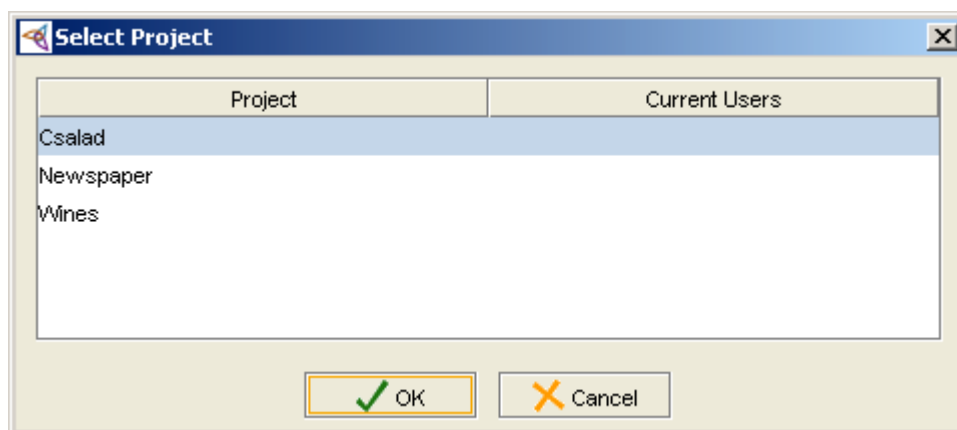
## 5. A kapcsolat felépítése a kliensek és a szerver között.

A szerver elindítása és a kliens telepítése után már csak a rendszer kipróbálása maradt hátra. A File/Open Project... menüt válasszuk ki, majd a megjelenő dialógus ablakon nyomjuk meg a Server feliratú gombot. A Host Machine Name (kiszolgáló számítógép neve) alá írjuk be a **dideki.tmit.bme.hu** hostnevet. A User Name és a Password sorokba a metaproject.pprj-ben adminisztrált és számunkra kiosztott belépési azonosító adatokat adjuk meg.



The image shows a dialog box titled "Open Project". On the left, there are three tabs: "File", "URL", and "Server". The "Server" tab is selected. The main area contains three input fields: "Host Machine Name" with the text "dideki.tmit.bme.hu", "User Name" with the text "protege", and "Password" with the text "\*\*\*\*\*". At the bottom, there are two buttons: "OK" with a green checkmark and "Cancel" with an orange X.

Az OK nyomógomb megnyomása után rövid időn belül be kell jelentkezzen a szerver az alábbi ablakkal.



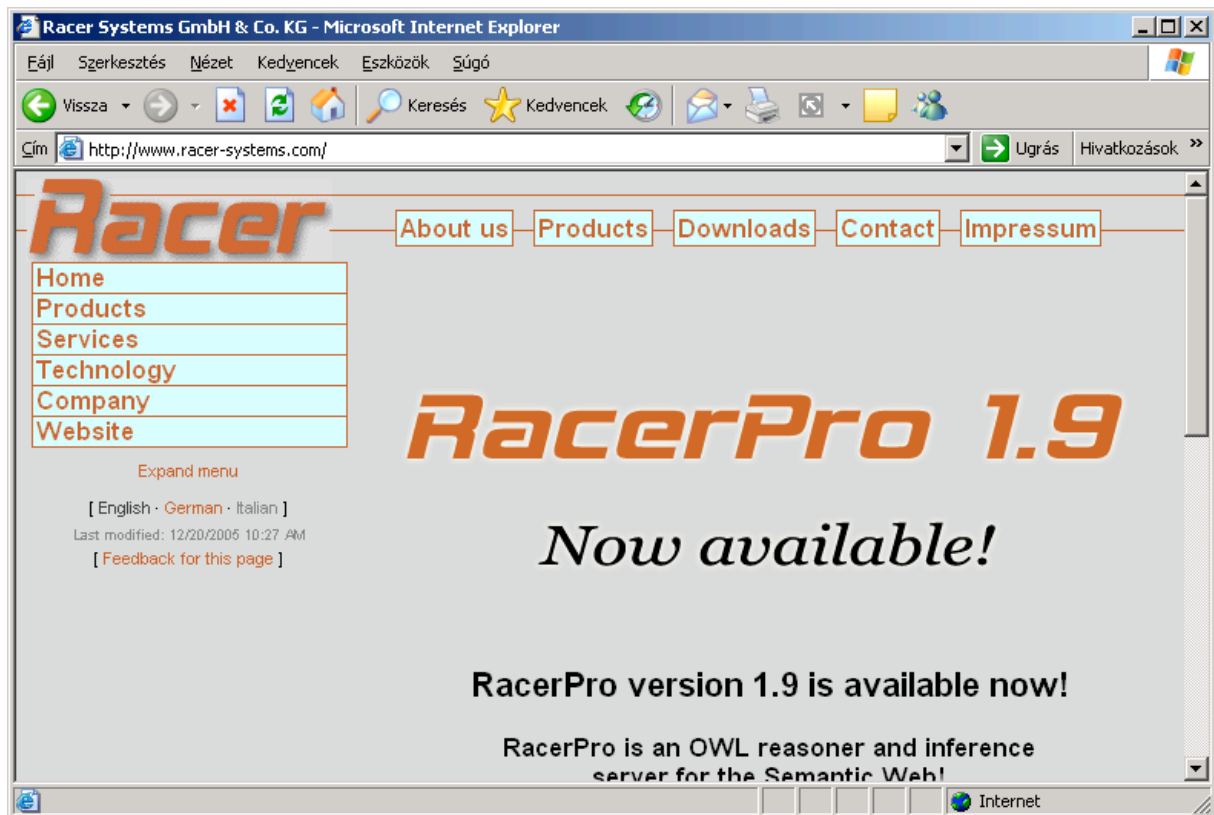
The image shows a dialog box titled "Select Project". It contains a table with two columns: "Project" and "Current Users". The "Project" column lists "Csalad", "Newspaper", and "Wines". The "Current Users" column is empty. At the bottom, there are two buttons: "OK" with a green checkmark and "Cancel" with an orange X.

A szerkeszteni kívánt projektet válasszuk ki és nyomjuk meg az OK nyomógombot. Ekkor a kliens letölti a szerverről az ontológia adatait és megjeleníti a képernyőn. Szerkesztés közben az anyagot elmenthetjük, kilépésnél ugyancsak visszairhatjuk a szerverre, és azon keresztül az adatbázisba az ontológiát.

## 6. A RacerPro 1.9. logikai következtető gép telepítése és használata

A Protégé rendszer külső leíró logikai következtető gép használatát preferálja. Ezek közül is teljesítményében kimagaslik a Racer Pro 1.9. verziójú szoftver. A rendszerhez a

<http://www.racer-systems.com> URL-címen lehet hozzájutni. A logikai következtető gép alapvetően licenrdíj fizetése mellett használható, azonban a felsőoktatásban dolgozók és tudományos kutatásban résztvevők egy levélváltás után ingyenes félévre szóló használati jogot kaphatnak, amely talán meg is újítható.

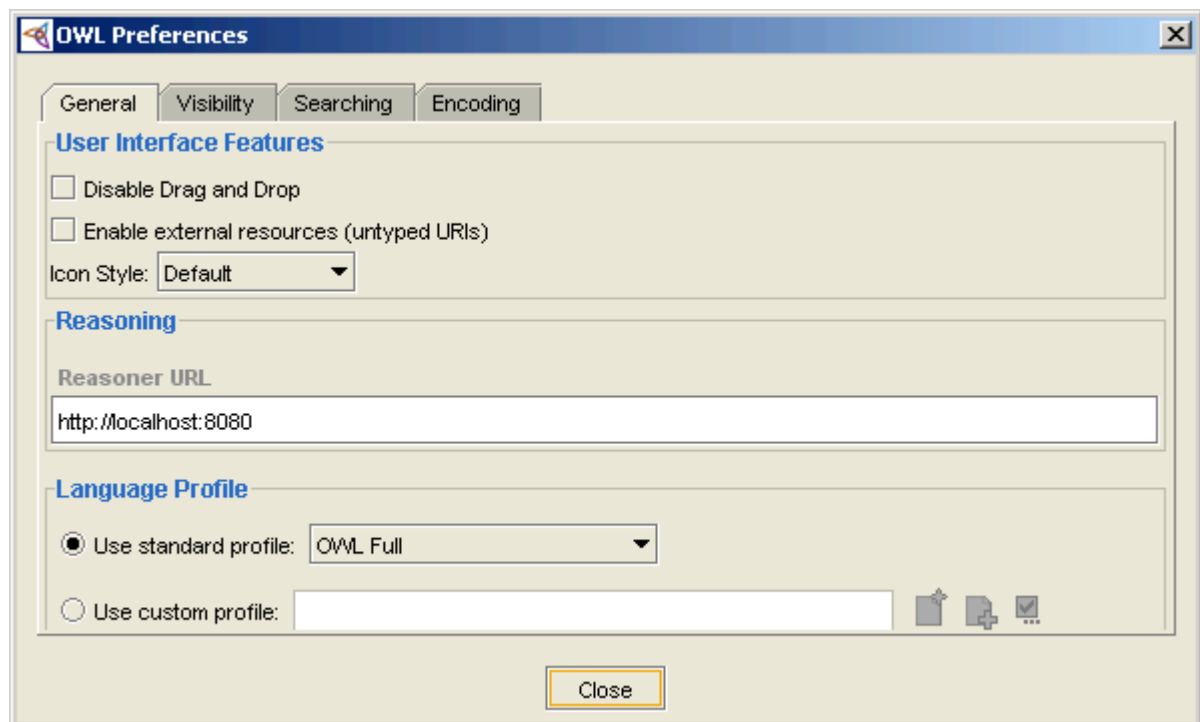


A telepítő anyag (RacerPro-1-9-0-Setup.exe) megszerzése után azt telepíteni kell a számítógépünkre. A levélben számunkra megküldött licenc fájlt (license.racerlicense) pedig be kell másolni a \Program Files\RacerPro-1-9-0\ alkönyvtárba. Ez után a következtető gépet el kell indítani a saját számítógépünkön.

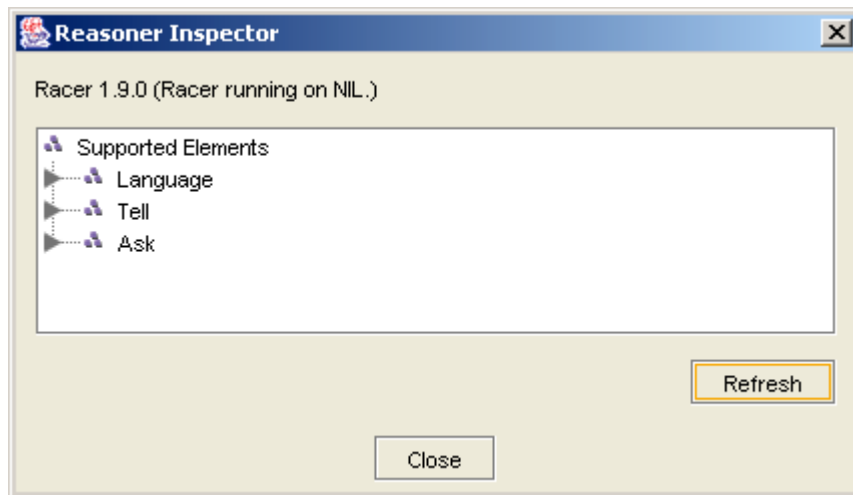
A RacerPro gép egy web szerverhez nagyon hasonló program, az Interneten érkező feladatokat oldja meg. Elvben távoli üzemmódra is alkalmas, azaz messziről érkező feladatokat is megold. A tesztelés során azonban nem szerette az átlapoltszekciókat, amikor két különböző feladaton szimultán kellett volna dolgoznia. Ezért az a legjobb, ha ki-kí a saját kliens gépén futtatja a RacerPro-t és a Protégé kliens ezt a logikai következtető gépet használja.

```
RacerPro
;;; =====
;;; Found license file
;;; E:\Program Files\RacerPro-1-9-0\license.racerlicense
;;; This copy of RacerPro is licensed to:
;;;
;;; Zoltan Alexin
;;; University of Szeged
;;; Arpad ter 2.
;;; Szeged
;;; 6720
;;; HU
;;;
;;; Initial license generated on 12-08-2005, 22:45 for 1.9.0.
;;; Desktop, Educational, on X86 Win32.
;;; This license is valid up to version 1.9.999.
;;; This license is valid from 12-08-2005, 00:00 to 04-30-2006, 23:59
;;;
;;; This timelimited demo license expires in 52 days, 7 hours and 25 min
;;;
;;; =====
HTTP service enabled for: http://localhost:8080/
TCP service enabled for: http://localhost:8088/
```

A következő gép szoftver adatait az OWL/Preferences... menüben tudjuk szerkeszteni.



A Reasoner URL sorba kell beírni a következő gép Internetes címét, URL-jét. Jelen esetben ez *gyárilag* be van állítva a <http://localhost:8080> címre. A RacerPro beállításai ezek (a konzol ablakban is ez jelenik meg), tehát ezen nem szükséges változtatni. A kapcsolat meglétét az OWL/Reasoner Inspector... menüvel tesztelhetjük le.



A Refresh gombbal frissíthetjük a képernyőt, és a RacerPro 1.9.0 be kell jelentkezzen. A támogatott protokoll elemeket megtekinthetjük a jobbra nyilakra kattintással. A következtető gép működését követhetjük például azzal, ha az OWL/Compute Inferred Types... menüt kiválasztjuk, amire az eredetileg személyként bevitt (Person) objektumpéldányok valódi hovatartozását kiszámítja (olyan lezárt osztályok esetén, amikor az osztályhoz társított logikai feltételekről kinyilvánítottuk, hogy azok pontosan – akkor és csak akkor – szinten meghatározzák hogy az adott osztályba mely objektum példányok tartoznak). A lezárt osztályokat a kis sárga körbe írt három vízszintes vonal jelzi. A melléjük írt szám pedig az asserted/inferred (kézzel létrehozott/kikövetkeztetett) példányok számát adja meg.

