Banner Moodle Integration Public Release

Introduction

This code is what can be used to interface enrollments from Banner to your Moodle instance. The code contained below is hereby released under the GNU General Public License. Minor customization of the code may be necessary for your environment. This will create a set of views that can be accessed using the Moodle enrollments external Database configuration settings.

The environment this was developed on was Banner 8 (Oracle 11g) and Moodle 2.2.x. There is no reason this will not work with other combinations. Once you grasp the concepts of how the integration works, Banner programmers should easily be able to adjust the views to customize to your individual needs.

Banner itself is a Higher Education Software and is licensed from Ellucian. For more information about Ellucian, it's software offerings and copyrights, please visit their website at http://www.ellucian.com/

Moodle is an open source Course Management System released under similar Public License. For more information about Moodle, please visit their website at http://moodle.org.

Oracle is a commercial database package and you can learn more information about their products by visiting their website at http://www.oracle.com.

About the Authors

- Scott May - Assistant Director Computing Services. Scott started with the State University of New York in 1994 as a Banner programmer. He has been at Delhi since 1996 and is the system and network administrator responsible for the underlying technologies supporting all campus computing systems. He's a strong Moodle advocate based on it's reliability, support network, excellent programmatic design and cost efficiencies.
- Grady Miller - Manager of Instructional Technology. Grady started with the State University of New York in 2009 as the Assistant Coordinator of Online Instruction. Also, he was a Banner programmer in a past capacity. Currently he is the primary Moodle administrator on campus and regularly works with faculty with regard to its use.

Intended Audience

This document is targeted to Moodle Administrators and Banner DBAs or programmers.

What is missing?

- This code works for us, but it may require minor adjustments based on your business practices. The database view syntax is written in Oracle (11g) and it would be necessary to change if your database environment for Banner is different.
- The other item which is missing would be information regarding firewall or protections you may have between your servers. This will require systems and network administrators to possibly assist in the implementation. The Moodle application server will need to be able to talk to the Oracle Database Server (for example).
- User directory integration between Banner and Moodle are required to make this work. Our implementation uses a secure LDAP implementation with a popular domain structure. Moodle allows many options, from LDAP, CAS, External database and on and on. The external database could be used to connect directly to Oracle itself if you so choose. We chose the popular domain structure because we have multiple servers and it would be available even during Banner maintenance periods. For enrollment we use SPRIDEN_ID from Banner which means your authentication source should also contain an ID number which aligns with that same identification.

Step 1 - Create moodleuser in Oracle

We created a dedicated owner for all the database objects, "MOODLEUSR". This user will own the views and will need to have access the tables necessary to compile that view within Banner. You will likely want to create a second user account that is just able to access the views from Moodle. For this we will keep a single user for simplicity of discussion and development. You should consult your Oracle DBA and organizations best practices for implementing security between.

Step 2 - Create the moodleuser.Category Table

Moodle uses categories for each course. In our instance we usually have term based categories like "Spring 2012" or "Fall 2013". The category table is just going to be the cross reference table we use. Your courses will show up under the default Miscellaneous category if you do not have a mapping in this table.

MOODLEUSR.CATEGORY TABLE CREATION CODE

```
CREATE TABLE IF NOT EXISTS "MOODLEUSR"."CATEGORY"
  (
    "TERM_CODE" VARCHAR2(6 BYTE) NOT NULL ENABLE,
    "ID"        VARCHAR2(2 BYTE) NOT NULL ENABLE,
    CONSTRAINT "CATAGORY_PK" PRIMARY KEY ("TERM_CODE") USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
STATISTICS STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1
FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT) TABLESPACE "DEVELOPMENT"
ENABLE
  )
  SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING STORAGE
  (
    INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT
  )
  TABLESPACE "DEVELOPMENT" ;
```

Category Codes

Category codes from the Moodle side can be found in table mdl_course_categories.There are two fields, ID and Name. We are going to want to map those to the appropriate term codes.

Note: The Moodle database table used in the example was created with the "mdl" pre-fix, your tables may vary in name.

Term Codes

Banner programmers will be familiar with the term code table STVTERM. We will be matching the stvterm.stvterm_code (in Banner) with mdl_course_categories.id (in Moodle).

The Moodle category IDs can also be identified from the URL for the category in Moodle. Navigate to a specific category and the URL will look something like this: "https://your_school.edu/course/category.php?id=19". The ID in this case is 19. The categories can be found under Site Administration > Courses > Add/edit courses.

Example Data

So we construct a table of data we wish to insert into the table. Here is an example of what our term_codes mapped to for our category codes.

Term_Code	ID
201201	4
201202	8
201209	10

Now we would just insert those values into the table through a SQL editor.

Insert Data into moodleuser.category

```
insert into moodleusr.category Values ('201201','4');
insert into moodleusr.category Values ('201202','8');
insert into moodleusr.category Values ('201209','10');
commit;
```

Step 3 - Creating the Course View

Now that we've created a category id to term code mapping, our next step will be to create a view that contains the course data we wish to make available to Moodle for course creation. For the Moodle enrollment to work we will need to have these 5 pieces of data:

1. COURSE_ID - we create this unique identifier by combining the Banner 5 digit CRN followed by the 6 digit term code. Example: 10001201209
2. SHORTNAME - we create this value by using the subject code, course number, CRN and term code. Example: MATH-128-11351-201302

3. LONGNAME - we create this value by using the subject code, course number, crn, course title and term description. Example: MATH-128-11351-College Algebra (Spring 2013)
4. CATEGORY - this will be course term code translated using the moodleusr.category table created in step 1 into the Moodle category ID.
5. START_DATE - this will be populated by the date in ssbsect_ptrm_start_date, in lay-mans terms, the course section start date. (Note: This information is not being used by Moodle currently, but is included in in the view in anticipation of further Moodle development and the inclusion of the start date field to the Moodle enrollment synced fields for course creation.)

Create MOODLEUSR.COURSE View

```
CREATE OR REPLACE FORCE VIEW "MOODLEUSR"."COURSE" ("COURSE_ID", "SHORTNAME", "LONGNAME", "CATEGORY",
"START_DATE")
AS
  SELECT ssbsect_crn
    ||ssbsect_term_code,
    ssbsect_subj_code
    ||'-'
    ||ssbsect_crse_numb
    ||'-'
    || ssbsect_crn
    ||'-'
    ||ssbsect_term_code,
    ssbsect_subj_code
    ||'-'
    ||ssbsect_crse_numb
    ||'-'
    || ssbsect_crn
    ||'-'
    ||NVL(ssbsect_crse_title,a.scbcrse_title)
    || ' ('
    ||stvterm_desc
    ||')',
    cat.id,
        ssbsect_ptrm_start_date
  FROM stvterm,
    ssbsect,
    scbcrse a,
    moodleusr.category cat
  WHERE stvterm_code      = cat.term_code
  AND ssbsect_term_code   = stvterm_code
  AND a.scbcrse_subj_code = ssbsect_subj_code
  AND a.scbcrse_crse_numb = ssbsect_crse_numb
  AND a.scbcrse_eff_term  =
    (SELECT MAX(b.scbcrse_eff_term)
    FROM scbcrse b
    WHERE b.scbcrse_subj_code = ssbsect_subj_code
    AND b.scbcrse_crse_numb   = ssbsect_crse_numb
    AND b.scbcrse_eff_term   <= ssbsect_term_code
    );
```

Step 4 - Creating the Enrollment View

Thus far we've created the foundation piece to supply courses to Moodle. We cannot create enrollments into Moodle until the courses exist. So our next step is to create the enrollment view which will provide us that information.

Customizations - You will almost certainly need to make adjustments here, so please read carefully.

1. Registration Codes - You will need all your sfrstcr_rsts_code values that you want enrolled. "RE", "RW" are Banner defaults, yours may be the same different or more numerous.
2. Course Status Codes - You will need to verify your ssbsect_ssts_code value. "A" appears to be a Banner default value for an "Active" course.
3. Date Range - Modify the the date/term logic to suit your needs. As written the view provides enrollments for courses whose start date date is within a range of 12 months prior to 6 months in the future of the time the query is executed.

4. Moodle Role - Moodle expects us to provide 2 things, a Course ID, a user ID and a role. To keep local terminology consistent we do not use the default Moodle values in this example. The roles are can be found in the Moodle > Site Administration > Users > Permissions > Define Roles. You should use the "Short name" in your view and respect all case sensitivity. We use 'student' and 'instructor', replace with your desired respective roles.

Create View moodleusr.enrollment

```
CREATE OR REPLACE FORCE VIEW "MOODLEUSR"."ENROLLMENT" ("COURSE_ID", "ID", "ROLE")
AS
  SELECT ssbsect_crn
    ||ssbsect_term_code,
    spriden_id,
    'student'
  FROM stvterm,
    ssbsect,
    sfrstcr,
    spriden
  WHERE (stvterm_start_date BETWEEN TRUNC(SYSDATE) AND TRUNC(ADD_MONTHS(SYSDATE,6))
  OR TRUNC(SYSDATE) BETWEEN stvterm_start_date AND ADD_MONTHS(stvterm_end_date,12))
  AND ssbsect_term_code   = stvterm_code
  AND ssbsect_ssts_code   = 'A'
  AND sfrstcr_term_code   = ssbsect_term_code
  AND sfrstcr_crn         = ssbsect_crn
  AND sfrstcr_rsts_code  IN ('RE','RW')
  AND spriden_pidm        = sfrstcr_pidm
  AND spriden_change_ind IS NULL
  UNION
  SELECT ssbsect_crn
    ||ssbsect_term_code,
    spriden_id,
    'instructor'
  FROM stvterm,
    ssbsect,
    sirasgn,
    spriden
  WHERE (stvterm_start_date BETWEEN TRUNC(SYSDATE) AND TRUNC(ADD_MONTHS(SYSDATE,6))
  OR TRUNC(SYSDATE) BETWEEN stvterm_start_date AND ADD_MONTHS(stvterm_end_date,12))
  AND ssbsect_term_code   = stvterm_code
  AND ssbsect_ssts_code   = 'A'
  AND sirasgn_term_code   = ssbsect_term_code
  AND sirasgn_crn         = ssbsect_crn
  AND spriden_pidm        = sirasgn_pidm
  AND spriden_change_ind IS NULL;
```

Example Output from a query

COURSE_ID	ID	ROLE
10001201209	123456789	student
10002201209	987654321	instructor
...	...	...

Step 5 - Configure Moodle Enrollment for External Database

This is where it becomes difficult to provide specific instruction. However I will proved highlights and links to Moodle Documentation which should assist in completing the implementation.

1. Managing Authentication - You need to be able to link your authenticated user IDs with the Banner IDs creating their enrollments. You will want to have a Cron or scheduled task running to make sure your users are synchronized. Enrollments will occur and process every time a user logs in, however we often find pre-semester faculty reported no students in the classes. By synchronizing all users with Moodle, all enrollments will show correctly and accurately. Moodle provides scripts for example under the \auth\ldap\cli is sync_users.php used for LDAP recurring synchronization tasks. (Site Administration > Plugins > Authentication > Manage Authentication)
2. Managing Enrollments - (Site Administration > Plugins > Enrollments > Manage enrol plugins)
    a. Under this you will want to modify the External Database Settings.
    b. Database connection - this is where you will enter all your data to connect to your Oracle server. You will need server FQDN or IP address, your moodleusr credentials, the database name. We do check the use sybase quotes for connecting to Oracle and enable Debug to assist you with your testing.

External database connection

Database driver
enrol_database | dbtype

oci8 Default: Empty

ADOdb database driver name, type of the external database engine.

Database host
enrol_database | dbhost

oracleserver.domain.edu:1521 Default: localhost

Type database server IP address or host name

Database user
enrol_database | dbuser

MOODLEUSR Default: Empty

Database password
enrol_database | dbpass

•••••••• Unmask

Database name
enrol_database | dbname

devl.oracleserver.domain.edu Default: Empty

Database encoding
enrol_database | dbencoding

utf-8 Default: utf-8

Database setup command
enrol_database | dbsetupsql

Default: Empty

SQL command for special database setup, often used to setup communication encoding - example for MySQL and PostgreSQL: *SET NAMES 'utf8'*

Use sybase quotes
enrol_database | dbsybasequoting

Default: No

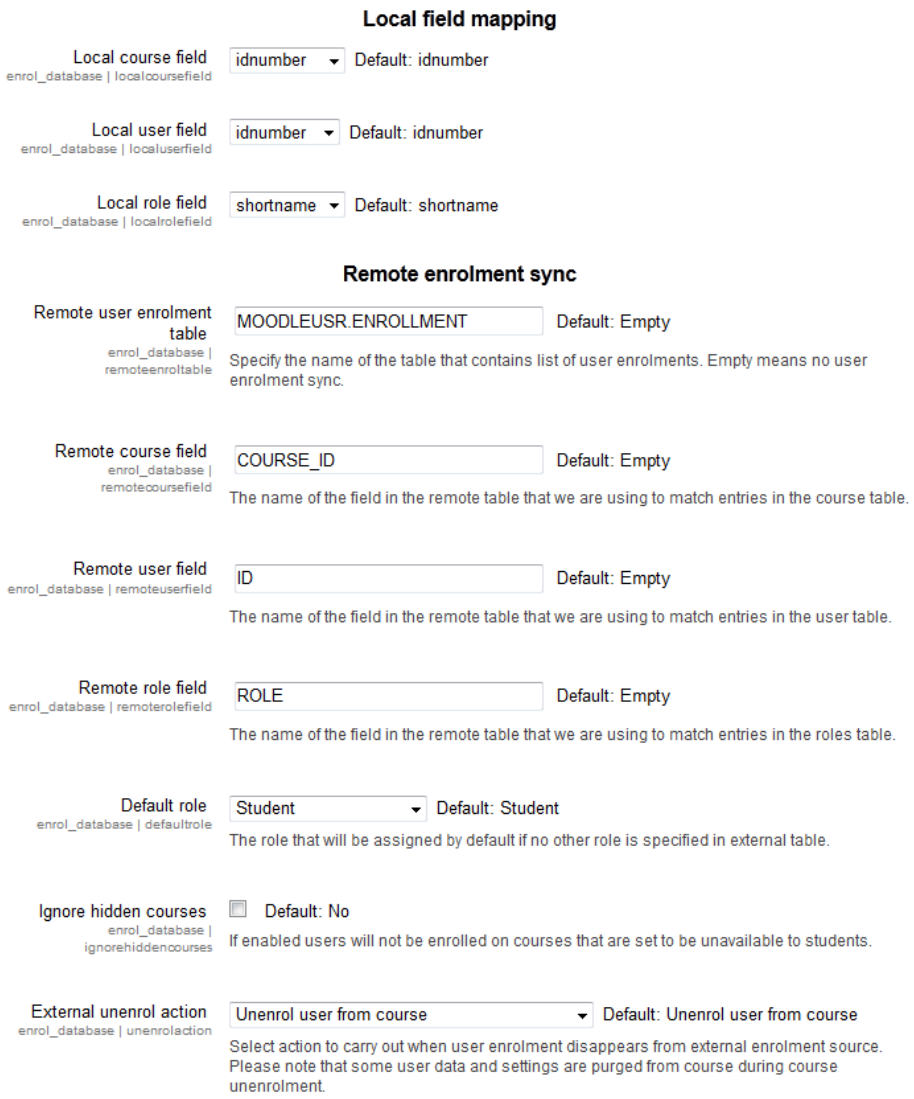
Sybase style single quote escaping - needed for Oracle, MS SQL and some other databases. Do not use for MySQL!

Debug ADOdb
enrol_database | debugdb

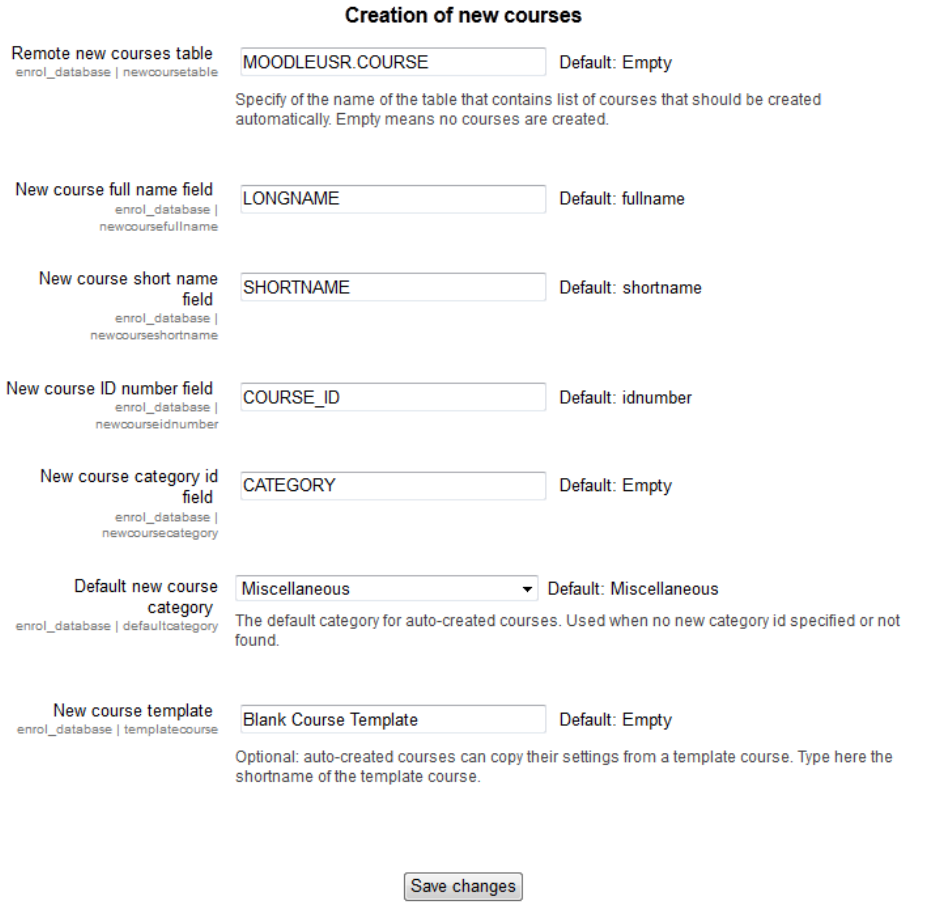
Default: No

Debug ADOdb connection to external database - use when getting empty page during login. Not suitable for production sites!

c. Next we configure the local field mapping, which we have not changed. The second part is the Remote enrolment sync which is where we are going to enter the specific data about our newly created views. The biggest choice here is how to handle unenrolled users and we just unenroll them from the course.



d. The final part of the external database configuration is to enter the creation of new courses. The first 5 items are where we enter our view configuration to be selected from the database. The second two you may set to you own preference. Note if you have missed a category mapping in Step 2, you may find many courses appearing in your Default new course category. You may wish to create a special category to catch these.



e. Save your settings.

Step 6 - Test and Schedule Your Enrollment Sync

From our moodle root folder on the moodle server you will find a folder ..\enrol\database\cli\sync.php. The first couple of times you run this launch it from an interactive command line interface so you can monitor the results. You may want to run it a couple times to make sure. When you are satisfied your process is completing as desired, set up a recurring task on the Moodle server to run the sync at your desired interval. We would recommend at least daily. In our environment we run a user sync, and then 10 minutes later our enrollment sync in separate jobs. We do this so that a failure of one task doesn't impact the other task from completing. Mostly personal preference, but we don't want a user sync failure to stop enrollment sync and vice versa. This job pair runs once hourly, 24/7/365.

Step 7 - Sit Back and Relax

Now that your Banner to Moodle integration is up and running it's time to take 5 minutes, reflect on your success and have a nice cold glass of water. Now get back to work, there's always more to be done!