Roodle 2.0 – Customization: CACE - Course Auto-Create & Enrol

Contents

Roodle 2.0 – Customization: CACE - Course Auto-Create & Enrol

Introduction

This middleware solution manages course creation and enrolment in Roodle (RRU customized Moodle) based on data in the Agresso Student system. The middleware is written in PHP/MySQL and utilizes some of the Moodle libraries to minimize code duplication while maintaining Moodle's business rules. PHP/MySQL solution was chosen as it allows complete customization capability as well as a large base of people who can maintain the code.

Installation

1. MS SQL Server setup – to be performed by DBA
    - a. In the Agresso database (VSS: $/AIS_Data/DB_Dev/SQL_2005/Agresso/views/IS_DataMart):
        - a. Execute uvw_DataMart_StudentEnrollment.sql
        - b. Execute uvw_DataMart_Learn_NewCourseOfferings.sql
    - b. In the IS_Datamart database (VSS: $/AIS_Data/DB_Dev/SQL_2005/IS_DataMart/procedures/mgmt_scripts):
        - a. Execute Moodle-CACE tables and security.sql
    - c. In the IS_Datamart database (VSS: $/AIS_Data/DB_Dev/SQL_2005/IS_DataMart/procedures):
        - a. Execute usp_Update_Agresso_StudentEnrollment.sql
        - b. Execute usp_Update_Agresso_Learn_NewCourseOfferings.sql
    - d. In the IS_Datamart database (VSS: $/AIS_Data/DB_Dev/SQL_2005/IS_DataMart/procedures/Moodle):
        - a. Execute Learn_usp_GetNewCourses.sql
        - b. Execute Learn_usp_GetStudentEnrolments.sql
    - e. Set up job to synch data between Agresso and IS_DataMart

2. Clone the CACE project from GitHub
    - a. Navigate to Moodle root in shell.
    - b. `cd local`
    - c. `git clone https://github.com/royalroads/cace.git`

3. Rename or copy the cace_config.example.php file to cace_config.php and edit to reflect appropriate settings for the server
    - a. Set connection settings for the MS SQL database server

      (**Get this information from your DBA**)

        1. mssqlserver : the name of the MS SQL Server on which the database is stored
        2. mssqldb : the name of the database where the stored procedures are stored
        3. mssqluser : the name of the database user that will be used to perform the connection to the database

4. mssqlpwd : the password of the database user that will be used to perform the connection to the database

b. Set Miscellaneous CACE settings
   a. emailto : CACE will generate email notifications when courses are created. They go to this address.
   b. defaultcatid : default category for courses if destination category is not found
   c. log : absolute path to log file for CACE including the filename
   d. monthsahead : how many months into the future to look for new courses
   e. res_name : name of the default page that is added to each new course shell
   f. res_intro : description of the default page that is added to each new course shell
   g. default_courses : course id's for default courses
   h. admin_user : user id of the administrator account

4. Load course categories to create category structure
   The course category structure specified for Moodle 2.0 is pulls Program Type, Department, and Program from the Agresso database to build the structure in Moodle.
   a. Set up an ODBC MySQL connector – to be performed by DBA
      1. If it doesn't already exist, set up an ODBC MySQL connector to allow the creation of a linked serve between MS SQL and MySQL servers.
      2. Create linked server between MS SQL and MySQL servers
   b. Create/Load the temporary table in Moodle with the data – to be performed by DBA
      1. Execute `SQLJob_Deploy_mdlfinchCategorySource.sql (VSS: $/AIS_Data/DB_Dev/SQL_2005/Agresso/mgmt_scripts/CACE/)`
      2. Confirm the data is present by running a query against the table in MySQL client

```
SELECT * FROM mdl_tmp_category_source
```

   c. Create the course categories in Moodle
      1. Open a browser and navigate to `http://`*`moodle root`*`/local/cace/setup/caceloadcategories.php`
      2. The screen should output progress information as it executes
      3. Upon completion, copy the page results to a text file and store for installation confirmation purposes – call the file something like '`Moodle 20 Course Category population.log`'.
   d. Clean up – to be performed by DBA
      1. Drop the `mdl_tmp_category_source` table from the Moodle database

5. Install the CACE plug-in
   a. Log into Moodle using an administrator account
   b. The "upgrade" screen will appear or go to the Site administration -> Notifications to trigger the "upgrade" screen
   c. Click the "Upgrade" button and a screen with "local_cace Success" should appear
   d. Navigate to Site administration -> Plugins -> Local plugins -> Manage local plugins
      a. Should see something like - cace | 2012010901

6. Set up cron jobs
   Data does not come directly from Agresso, but is instead imported into IS_Datamart via a SQL Server Job every 30 minutes between 5:30 AM and 11:00 PM each day.

   a. Auto-create course shell
      Add a line to the crontab so that run_cace-createcourses.php is run once a week on Monday at 6:30 am :
      ```
      0 6 * * 1 wget -q -O /dev/null http://hostname/moodle/local/cace/run_cace-
      createcourses.php
      ```

   b. Auto-update course shell
      Add a line to the crontab so that run_cace-updatecourses.php is run Monday and Friday at 6:00 am (Needs to be called BEFORE run_cace-createcourses.php):
      ```
      30 6 * * 1,5 wget -q -O /dev/null http://hostname/moodle/local/cace/run_cace-
      upatecourses.php
      ```

   c. Auto-enrol students
      Add a line to the crontab so that run_cace-enrollstudents.php is every day at 10 am, 2 pm, and 5 pm :
      ```
      0 10,14,17 * * * wget -q -O /dev/null http://hostname/moodle/local/cace/run_cace-
      enrollstudents.php
      ```

Auto-create Course Shell

Creates new course shells in Moodle with basic information

Use cases

1. A new undergraduate course is created in the Agresso Student Information System that has a start date within 9 months of the current date
2. A new graduate course is created in the Agresso Student Information System that has a start date within 9 months of the current date
3. A new Credit-Non-credit course is created in the Agresso Student Information System that has a start date within 9 months of the current date
4. A new course is created in the Agresso Student Information System that has a start date within 9 months of the current date and is part of more than one program within the same department
5. A new course is created in the Agresso Student Information System that has a start date within 9 months of the current date and is part of more than one program within the across different departments

Algorithms / Code changes

Database changes

- `Agresso.dbo.uvw_DataMart_Learn_NewCourseOfferings`
- `IS_Datamart.dbo.Agresso_Learn_NewCourseOfferings`
- `IS_Datamart.Learn.usp_GetNewCourses`

- `Moodle.cace_newcourses table`

Code changes

CACE files are stored in the /local/cace folder

- `/db/install.xml` – db tables to be installed
- `/db/install.php` – any install related tasks
- `cace_config.php` – stores configuration information specific to CACE middleware
- `cacelib.php` – caceware specific functions
- `database.php` – database specific functions
- `mssql.php` – Microsoft SQL Server specific functions
- `run_cace-createcourses.php` – simple php script to allow the CRON to run CACE
- `dev_notes.txt` – stores content for the Development Notes resource.
- `version.php` – Moodle version management file

Auto-update Course Shell

Updates existing Moodle courses that have not started based on changes from Agresso.

Use cases

1. The start date of an undergraduate course is updated to a future, but not more than 9 months into the future
2. The description of a course offering is changed for a course that has a start date no more than 9 months into the future

Algorithms / Code changes

Database changes

- `Agresso.dbo.uvw_DataMart_Learn_NewCourseOfferings`
- `IS_Datamart.dbo.Agresso_Learn_NewCourseOfferings`
- `IS_Datamart.Learn.usp_GetNewCourses`
- `Moodle.cace_newcourses table`

Code changes

CACE files are stored in the /local/cace folder

- `/db/install.xml` – db tables to be installed
- `/db/install.php` – any install related tasks
- `cace_config.php` – stores CACE middleware settings
- `cacelib.php` – caceware specific functions
- `database.php` – database specific functions
- `mssql.php` – Microsoft SQL Server specific functions
- `run_cace-updatecourses.php` – simple php script to allow the CRON to run CACE

- `version.php` – Moodle version management file

Auto-enrolments – Students

Enroll/Withdraw students into appropriate courses in Moodle based on Agresso Student System enrolments

Use cases

1. A student is enrolled in a course that has not started and exists in Moodle
2. A student goes on leave (and is withdrawn from their courses
3. A student withdraws from a course that has not started and exists in Moodle
4. A student withdraws from a course that has started and exists in Moodle

Algorithms / Code changes

Database changes

- `Agresso.dbo.uvw_DataMart_Learn_GetStudentEnrolments`
- `IS_Datamart.dbo.Agresso_StudentEnrollment`
- `IS_Datamart.Learn.usp_GetStudentEnrolments`
- `Moodle.mdl_cace_enrolments`

Code changes

CACE files are stored in the /local/cace folder

- `/db/install.xml` – db tables to be installed
- `/db/install.php` – any install related tasks
- `cace_config.php` – stores CACE middleware settings
- `cacelib.php` – caceware specific functions
- `database.php` – database specific functions
- `mssql.php` – Microsoft SQL Server specific functions
- `run_cace-enrollstudents.php` – simple php script to allow the CRON to run CACE
- `version.php` – Moodle version management file

Testing

Note: Requires access to Agresso client. Knowledge of Agresso is an asset to understanding the following tests.

Test Case setup

In the Agresso client (alpha or beta)

1. Create a new course offering that has a start date within the next 9 months
    a. Add the course to a program offering structure that will start in the near future
    b. Make note of the following:
        i. Course title
        ii. Course Start date
        iii. Course Level
        iv. Program code/Program Offering Start date
        v. Ed depart code (on the Relation tab in the Reporting folder of the Program template)
        vi. Delivery mode (Program offering)

2. Update the start date and description of an existing course offering that has already been imported into Moodle
    a. Open a course offering and update the offering description (offering_desc) and the start date so it one month into the future
    b. Make note of the following:
        i. Course title
        ii. Course Start date
        iii. Course Level
        iv. Program code/Program Offering Start date
        v. Ed depart code (on the Relation tab in the Reporting folder of the Program template)
        vi. Delivery mode (Program offering)

3. Create a new "shared" course offering that is shared between 2 programs that have the same department.
    a. This is accomplished by including the course in the Program Offering structure
    b. Make note of the programs and department

4. Create a new "shared" course offering that is shared between 2 programs that have different departments.
    a. This is accomplished by including the course in 2 different Program Offering structures
    b. Make note of the programs and departments

5. Enrol one or more students in the new course (the student user account must exist in Moodle and have a matching student_pk in the id field on their profile)

   a. Go to the Goal of any registered student and click on the Enroll tab
   b. Click on the "Insert row" button to add a new enrolment (a new screen will open)
   c. Enter the Course ID and Course Offering ID for the course that was created in step 1 – press the "Tab" key to pull in the relevant info and save the new Enrolment.

6. Drop a student from a course that has already been imported into Moodle
   a. Go to the Service Offering screen and enter the Service ID and Offering ID and press the "Tab" key to bring up the course
   b. Go to the Enroll tab, locate the student and double-click on their record
   c. The student's enrolment should appear.
   d. Select "Drop" from the "Enrollment closure" drop-down menu to pull up the Drop screen
      i. Enter today's date in the Closure date and Eff. Refund date
      ii. Press the "F9" key to open the Closure tree and pick a value from the "Withdrawl, Deferral or Discontinuation" item
   e. Save the Drop

Test Cases

1. New course
   a. Run the ***run_cace-createcourses.php*** to create new courses that have been created in Agresso
   In Moodle, the new course that was created in Agresso (Step 1 of Test Case setup) should:
   - exist under the correct program category
   - has a "Development Notes" page resource included that opens in the current page
   - confirm course settings
     - have the correct course full name format
     CourseCode [*ProgramCode ProgramStartYear-Section DeliveryMode*] *course start date – course description*
     - have the correct course short name format
     CourseCode [*ProgramCode ProgramStartYear-Section DeliveryMode*] *course start date*
     - have Course ID number of "*course code__course offering code*"
     - have Format = "Topics format"
     - have Course start date the same as Agresso record
     - have Availability = "This course is not available to students"

2. New shared course (within same department)
   a. Handled by 1.a Run the "run_cace-createcourses.php" script
   The new course should have same confirmations as Test 1, but the Names should be different:
   confirm course settings
     - have the correct course full name format
     CourseCode [*DepartmentCode ProgramStartYear-Section DeliveryMode*] *course start date – course description*
     - have the correct course short name format
     CourseCode [*DepartmentCode ProgramStartYear-Section DeliveryMode*] *course start date*

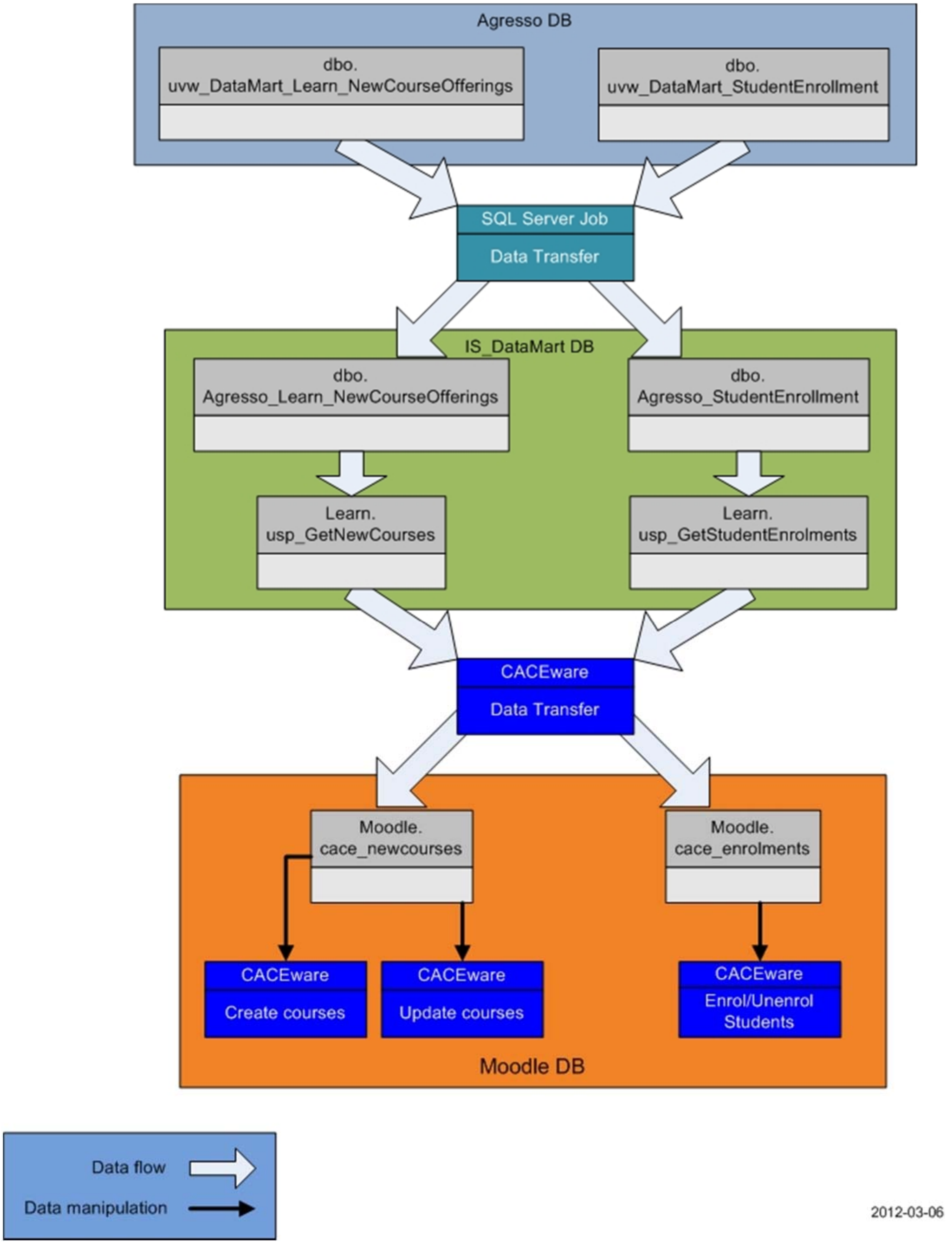
3. New shared course (across departments)
   a. Handled by 1.a Run the "run_cace-createcourses.php" script
   The new course should have same confirmations as Test 1, but the Names should be different:
   confirm course settings
   - have the correct course full name format
   CourseCode [*DepartmentCode ProgramStartYear-Section DeliveryMode*] *course start date – course description*
   - have the correct course short name format

CourseCode [*DepartmentCode ProgramStartYear-Section DeliveryMode*] *course start date*

4. Update existing course start date/description
    a. Run the ***run_cace-updatecourses.php*** to update the course in Moodle with appropriate data
    Confirm course settings:
    - have the correct course full name format
    CourseCode [*DepartmentCode ProgramStartYear-Section DeliveryMode*] *course new start date – new course description*
    - have the correct course short name format
    CourseCode [*DepartmentCode ProgramStartYear-Section DeliveryMode*] *course new start date*
    Course start date the same as Agresso course offering

5. New student enrolment
    a. Run the ***run_cace-enrollstudents.php*** to update enrolment of students in a course
    Confirm that:
    - The student in step 5 is able to access the course in which they were enroled through the Agresso system.
    - The student in step 6 is not able to access the course from which they were drop through the Agresso system.

Appendix A – CACE process flow diagram



2012-03-06