AMP Student Registration Impact Assessment

|  |  |
| --- | --- |
| 1. Document Information | |
| Document Name | AMP Student Registration Impact Assessment |
| Service | Moodle |
| Author | Alexis Nelson |
| Contributors | N/A |

|  |  |  |
| --- | --- | --- |
| 1. Document History | | |
| **Version** | **Date** | **Summary of change** |
| 0.1 | 09.04.2019 | Initial draft |

|  |
| --- |
| This document may include data that is **CONFIDENTIAL** and shall not be disclosed outside UCL and shall not be duplicated, used, or disclosed in whole or in part for any purpose other than to evaluate and implement procedures defined within this document. |

Contents

[i. Document Information 1](#_Toc13556354)

[ii. Document History 1](#_Toc13556355)

[1 Introduction 3](#_Toc13556356)

[2 Proposed changed 3](#_Toc13556357)

[2.1 Context 3](#_Toc13556358)

[2.2 The change 3](#_Toc13556359)

[3 Moodle impact 5](#_Toc13556360)

[3.1 Current state 5](#_Toc13556361)

[3.2 Remedial options 6](#_Toc13556362)

[4 Remedial action 7](#_Toc13556363)

# Introduction

This document outlines the impact on Moodle of the second tranche of changes being introduced this year (2019/20) to SITS as part of the Academic Model Project.

This change is to facilitate strictures around the student module registration process. The full change is more about ensuring the system enforces the policies around that process, however the technical change as it impacts Moodle is only in the way the data is stored. Different column, different values.

# Proposed changed

## Context

Currently, student module registration, while it has a reasonably well defined process, is not well controlled by the system. The idea is that the basic process should be as follows:

1. Student selects module, it is a new selection
   1. An SMS record is created, the status is NEW
2. The department which teaches that module approves the student to take that module, it is an approved selection
   1. The SMS record is updated to APP
3. The student’s parent department approves the student to take that module, it is a confirmed selection
   1. The SMS record is updated to CON
4. The student is now on that module
   1. An SMO is created

This is the “happy path” for module selection. Along the way either department – teaching or parent – could reject the module. It is also possible to reverse a confirmation. As confirming a module creates a new record, meaning both a module selection confirmation record and a module enrolment record exist at the same time. It is possible to delete either of those independently, as well as creating them independently. This means the data can be in a very indeterminate state – rejected but enrolled, confirmed but not enrolled. It has also been possible for departments to skip steps, meaning a parent department could confirm a selection the teaching department was yet to approve.

This change looks for add a more thorough workflow into the system, hopefully removing some of that ambiguity. As mentioned, this is done mostly through changes to the system as the user interacts with it. However, it does also bring about some changes to the way the data is stored.

## The change

Currently, as per the workflow overview above, module registration involves two tables and a series of statuses stored in a user defined field. While the tables won’t change, the statuses and the field in which they are stored will. The current statuses are simply (as above) NEW, APP, CON and REJ. They describe at a very high level what has happened to the module selection. The new set of statuses, in the new field, will hold a lot more detail of the selection:

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Name** | **Usage notes** | **CMIS Status** |
| C | Compulsory | Compulsory module (this can be used on either SMS or SME type elements to allow compulsory modules to be flagged for exam boards looking at modular flexible students) | CON |
| O | Optional | Optional modules (fully approved) | CON |
| E | Elective | Elective modules (fully approved) | CON |
| NC | Compulsory (non-condonable) | Used to identify non-condonable modules (derived from diet). All non-condonable modules must be compulsory. | CON |
| AO | Optional | Optional module (automatic parent department approval of module selections) (defined in diet) | CON |
| AE | Elective | Elective module (automatic parent department approval of module selections) (defined in diet) | CON |
| TO | Optional | Optional modules approved by teaching department only | APP |
| TE | Elective | Elective modules approved by teaching department only | APP |
| PO | Optional | Optional modules approved by parent department only | use APP |
| PE | Elective | Elective modules approved by parent department only | use APP |
| NO | Optional | Optional modules (not approved) | NEW |
| NE | Elective | Elective modules (not approved) | NEW |
| ~~S~~ | ~~Supersede~~ | ~~Superseded module (excluded from progression and award)~~ | - |
| ~~R~~ | ~~Retake~~ | ~~Retaken modules~~ | - |
| RO | Rejected | Rejected (Optional) | REJ |
| RE | Rejected | Rejected (Elective) | REJ |
| DO | Deleted | Deleted (Optional) | use REJ |
| DE | Deleted | Deleted (Elective) | use REJ |
| DC | Deleted | Deleted (Compulsory) | use REJ |
| DN | Deleted | Deleted (Non-condonable) | use REJ |

<https://wiki.ucl.ac.uk/display/AMP/Programme+Diet+Data#ProgrammeDietData-SelectionStatus(SES_CODE)>

The column CMIS status actually refers to the existing status. CMIS is just a known consumer. Moodle also uses the statuses in the same way.

# Moodle impact

## Current state

The main thing to note is that there is no plan to retrospectively change data. The new process and thus the new data will be created the next time enrolment happens. The AMP project will provide a timeline for this, indicating when the next round of enrolments will happen and thus when the impact will first be realised.

The current use of this data in Moodle is via a materialised view provided by SITS - UCL\_BDO\_MV\_MOODLE\_MOD\_REG – which lists student module registrations. The view has both the existing column, a mapping of SMS\_UDF9 to STATUS (which also does some transformation logic), and the to be column SES\_CODE. Only the STATUS column is used.

The status is also used in the Portico Enrolment mapping. There is an abstraction table (SITS\_REGISTRATON\_STATUSES) which allows the different statii to be compared to one another (by mapping them to numbers, REJ=0, NEW=1, APP=2, CON=3).

Currently, those mappings work thus:

|  |  |  |
| --- | --- | --- |
| Registration Status | ‘Level’ | Action |
| REJ | 0 | Not enrolled in Moodle, will be removed if currently enrolled |
| NEW | 1 | Enrolled if Portico Enrolment mapping in Moodle set to allow Pre-approve |
| APP | 2 | Enrolled in Portico |
| CON | 3 | Enrolled in Portico |

To wit:

1. Portico Enrolment mapping is created
   1. Module ECON0001 is selected, as Pre-Approval
   2. ECON0002 is selected, as Pre-Approval
   3. ECON0003 is selected as approved registrations only
2. This is stored in the database:

|  |
| --- |
| Mapping against course |
| ECON0001, NEW, Level 1 |
| ECON0002, NEW, Level 1 |
| ECON0003, APP, Level 2 |

1. When the MIM import runs, it compares the student’s registration status to the Portico Enrolment mapping via the numeric translation, for example, three students each have one of:
   1. Registration ECON0001 has had no approval (status = NEW)
   2. Registration ECON0002 is approved by the teaching department (status = APP)
   3. Registration ECON0003 has had no approval (status = NEW)

|  |  |
| --- | --- |
| Mapping against course | Registration status against student |
| ECON0001, NEW, Level 1 | ECON0001, NEW, Level 1 |
| ECON0002, NEW, Level 1 | ECON0002, APP, Level 2 |
| ECON0003, APP, Level 2 | ECON0003, NEW, Level 1 |

1. The student’s registration status is compared to that of the Portico Enrolment mapping, via the status level:
   1. Registration ECON0001-NEW has level 1, Mapping ECON0001-NEW has level 1, so that student is added to the Moodle course
   2. Registration ECON0002-APP has level 2, Mapping ECON0002-NEW has level 1, so that student is added to the Moodle course
   3. Registration ECON0003-NEW has level 1, Mapping ECON0003-APP has level 2, so that student is not added to the Moodle course.

|  |  |  |
| --- | --- | --- |
| Mapping against course | Registration status against student | Action |
| ECON0001, NEW, Level 1 | ECON0001, NEW, Level 1 | Enrol in course |
| ECON0002, NEW, Level 1 | ECON0002, APP, Level 2 | Enrol in course |
| ECON0003, APP, Level 2 | ECON0003, NEW, Level 1 | Do not enrol |

The issue then is that if the incoming statii from Moodle are changing from REJ, NEW, APP, CON and the column from which to take that value is changing, the import code, the Portico Enrolment block code and the data in SITS\_REGISTRATION\_STATUSES would potentially need to change or students will no long be enrolled on Moodle courses via their modules.

Incidentally, courses mapped by departments are also affected in the same way as department mappings use the same logic. There are only 47 courses mapped this way, though.

## Remedial options

There are two main ways to fix this issue. The first is to make changes outlined above, to the code and the database, the second to remap the data at source.

### Refactoring MIM/Portico Enrolment to use the new status codes

Changing those three areas are each individually small pieces of work. Repointing the import to use a different column only affects one file; changing the data in the table would be a couple of SQL statements and changing the Portico Enrolment block would similarly only take a moment (and may even improve it for future changes).

**Benefit**: Moodle remains consistent with the data in SITS and that data is arguably more immediately obvious when troubleshooting as it is represented in the Moodle database as it is in SITS.

**Downside**: SITS business logic is repeated in Moodle, rather than being abstracted away. It is also technically more work at the Moodle end, which introduces some risk, even if it is ultimately minimal.

### Remapping the data at source

Remapping the data at source is also technically simple. At the moment the STATUS column in the view is a composite of the UDF6 SMS status and the fact of an SMO existing – that is, if an SMO exists, the status Moodle gets is defacto CON (irrespective of UDF6), otherwise it is the value of UDF6. It would be reasonably trivial to change that and implement the logic above converting the various new statii to the existing ones.

**Benefit**: this particular bit of business logic is maintained at source rather than being distributed and open to drift as that logic changes. It also means fewer changes at the Moodle end (none, really), which is useful in containing the possibility of error. That change is also more aligned to a service oriented architecture

**Downside**: troubleshooting is made a little less transparent. It will also be difficult to divorce from the original terminology even if that loses its meaning as time goes on (or, at least, the old terminology will be kept current artificially). That is less of an issue if the business confirms that the old statuses are still valid. This does look to be the case.

It is also worth noting that changing the Portico Enrolment block to use levels rather than directly using module registration statuses is probably a worthwhile piece of refactoring anyway. However, it is only AMP work if it is borne of a change to those statuses.

**Outstanding**: Confirm from Registry what the logical states of a module selection can be and if they still ultimately just map back to the original four.

# Remedial action

Pending confirmation, the remapping data at source route seems the most sensible. The possibilities, then are that the current set of selection statuses can be kept, in which case no changes need to be in Moodle at all, or they should change, in which case the block and mapping table need to change.

Unlike the previous change, there is no immediate impact to this. The registration timeline to be provided by registry will cover the next time an impact could be felt. In any case it should be after the standard cut-off/rollover for Moodle courses.

As such the main steps are to change the code as necessary, run the tests in conjunction with AMP and then track the first change as they are due to filter through.

This impact document will be updated once more has been made clear from AMP.