Timetable Database Restructuring (Computational)

by

Patricia Noguera
2006-2007

Supervisor: Stephen Houston

Acknowledgements

I would like to express my gratitude to Dr. Paul Record for providing me with help when I needed it.

My special thanks to Professor Stephen Houston, who wisely oriented and guided me through my project.

To my family and friends, thanks for their support and for guiding my path.

Content

Introduction

Outline of project

The university is to restructure its academic sessions from 3 terms of 10 weeks to 2 semesters of 14 and 16 weeks. This means that current room timetable data will need to be converted into a new teaching pattern. It is also likely that all codes used to identify courses and modules will need to change to make a distinction between term and semester academic delivery.

The university uses a software package called Syllabus Plus for its timetabling. This package can perform scheduling functions however it is currently employed only as a room booking system at present.
In academic session 2008-2009 the university will be restructuring its academic year from 3 terms of 10 weeks to semesters of 14 weeks and therefore major changes will be required to the timetabling information. This project has two functions, both with practical and relevant applications to the timetabling of the university.

The timetable database will need to be amended to reflect the changes in the academic year. All module codes, course codes, course structures, active weeks will be changed to new codes and patterns.

The project has looked a number of aspects associated with the data held in a term format and conversion to a semester format. All of the changes can be done manually on Syllabus Plus but given that there are 1400 modules with activities associated with them and 8500 scheduled activities, a manual conversion would take a considerable amount of time.

This project will also identify any issues concerning current data format and potential problems with data conviction.

Discussion of Syllabus Plus software

The software package called Syllabus Plus has the ability to schedule taught activities based on student registration and room and staff requirements (called suitabilities and constraints). Presently the university timetable is organised manually relying on local knowledge and accumulated experience of timetabling at Heriot-Watt University. The software is used as a repository for room booking information. This information is on a shared database and each timetables has open access to objects in the database that belong to them and shared resources. A subject timetables has the authority to make or amend their subjects' activities and locate in shared locations. A subject timetable has limited permissions, but sufficient to ensure that they can reserve teaching accommodation (a resource) to allow their programmes to run.

The database is organised as follows: A programme of study is a collection of mandatory and optional modules that are taken by students on that course. Each module has a number of activities associated with it. These activities are the points of contact between a member of staff and the students. The activities can have different types (lecture, tutorial, lab etc) and can run for different durations. The university timetable is organised as having teaching between 9.15am and 6.15pm on Monday, Tuesday Wednesday Thursday and Friday. The day is split into 9, hourly slots. The software can have student registration information imported and can then be used to allocate students to activities and then make timetable decisions based on room availability, staff availability, room suitabilities, student number and student availability. (This is not used at present at Heriot-Watt University, all timetable decisions are taken using staff local knowledge.)

The timetable software is a relational database and stores all data in a 'house style'. The data can be exported as a text file for the purposes of 'rolling over' data from one academic session to another. The software works on a 52 week year. At the end of each year a new year is created by exporting the data and importing into a new database with the date of the first week amended. The export file is called an exchange file (.uef) and is structured text file.

Discussion of data format from software

The data format of file that Syllabus Plus program creates is structured always following the same configuration.

In the one hand, the file has declaration the objects which belong to database, these objects have an ID and their hostkey, so there is an Institution, 60 Departments, 50 Named Availabilities, 628 Programmes of Study, 14 Activity Types, one Students, 20 Suitabilities, 2066 Modules, 123 Student Sets, 277 Staff, one hundred and ninety Locations, 48 Activity Templates, 8763 Activities.

Every object is identified with the ID, so for example, the institution has the ID number 10000, departments have the Id number 27000, Named Availabilities have the ID 25000, Modules have the ID 11000, etc.

Bellow are examples of the uef file structure:

```
REM Declare 50 Named Availabilities
25000   #SPLUSB19023
25000   #SPLUSB19024
25000   #SPLUS837C50
25000   #SPLUS895D00
25000   #SPLUS57641F
25000   #SPLUS5763D6                    HostKey of each named availabilities
25000   #SPLUS9CD9FA
25000   #SPLUS3F53F3
25000   #SPLUS666CEC
25000   #SPLUS9CDA04
…..
```

```
REM Declare 2066 Modules
11000   342GM2
11000   153YP3
11000   342SM2
11000   343EM2
11000   341RZ3
11000   341RC3
11000   151CL3
11000   151CP3                          HostKey of module
11000   263AC3
11000   262AW3
11000   132TH3
11000   143ES3
11000   269TF3
11000   269TG3
11000   312CN3
11000   344SI1
……
```

```
REM Declare 14 Activity Types
31000  #SPLUSADDAF9
31000  LP
31000  LL
31000  LC
31000  -
31000  C
31000  T
31000  #SPLUSF6730C                    HostKey Activity Type
31000  AL
31000  #SPLUSADDAF8
31000  LD
31000  S
31000  W
31000  #SPLUS447692
REM
```

In this last case, there fourteen Activity Types are declared and the ID of it is 31000 with every one hostkey that after it will be necessary to use in every specific object. The modules are identified with Id 11000 and named availabilities with ID 25000.

The file has every object specified, so for example, the specification of a Module:

```
REM Module: C44SI1
11004   344SI1   C44SI1                                  Name of module
11005   344SI1   SPANISH CASE STUDIES A                  Description
11020   344SI1   34                                      Department Number
11036   344SI1   0
11021   344SI1   0                                       Student number
11027   344SI1   -all              0
11027   344SI1   -all              1
11027   344SI1   +          332M4  1
11030   344SI1   -all
11039   344SI1   0                                       Student number
11051   344SI1   1
11051   344SI1   0
11061   344SI1   0          68
11063   344SI1   11111111110000000000000000000000000000000000000000000000
11081   344SI1   -all       000000000000000000000000000000000000000000000000000000000000   0   0   0
11081   344SI1   +          000000000011111111111111111111111111111111111111111111111111   0   0   1095
11081   344SI1   +          000000000011111111111111111111111111111111111111111111111111   1   0   1095
11081   344SI1   +          000000000011111111111111111111111111111111111111111111111111   2   0   1095
11081   344SI1   +          000000000011111111111111111111111111111111111111111111111111   3   0   1095
11081   344SI1   +          000000000011111111111111111111111111111111111111111111111111   4   0   1095
11331   344SI1   -all
11331   344SI1   +
11332   344SI1
11333   344SI1
11334   344SI1                                  Availability 52 weeks of year
11335   344SI1
REM
```

The second column is the hostkey, as this module has been specified before in the Modules declarations, we must find: 11000 344SI1
In the first line 11004, we have the name of module, line 11005 has the module description, line 11020 has the number of department where the module belong, line 11021 and 11039 is the population number, line 11063 is the term, in this case this module belong to Term 1. This will be explained later. Line 11081 is opposite of 11063.

Activity specification:

REM Activity: A13BD1:AL:131

12004	3045	A13BD1:AL:131				*Name of module*		
12005	3045	A13BD1:AL:131 (15.3MM1:U:131)				*Description*		
12020	3045	15				*Department Number*		
12036	3045	0						
12021	3045	0				*Student number*		
12022	3045	+	153MM1 0			Module hostKey		
12028	3045	1						
12031	3045	-all						
12032	3045	-all						
12035	3045							
12039	3045	0				*Student number*		
12051	3045	1						
12051	3045	0						
12060	3045	60						
12061	3045	0	61					
12062	3045	9						
12063	3045	11111111100000000000000000000000000000000000000000						
12081	3045	-all	00000000000000000000000000000000000000000000000000			0	0	0
12081	3045	+	00000000011111111111111111111111111111111111111111			0	0	1095
12081	3045	+	00000000011111111111111111111111111111111111111111			1	0	1095
12081	3045	+	00000000011111111111111111111111111111111111111111			2	0	1095
12081	3045	+	00000000011111111111111111111111111111111111111111			3	0	1095
12081	3045	+	00000000011111111111111111111111111111111111111111			4	0	1095
12100	3045	2						
12101	3045	2	555					
12102	3045	2	555					
12106	3045	-all						
12106	3045	+	0	2	555			
12106	3045	+	1	2	555			
12106	3045	+	2	2	555			
12106	3045	+	3	2	555			
12106	3045	+	4	2	555			
12106	3045	+	5	2	555			
12106	3045	+	6	2	555			
12106	3045	+	7	2	555			
12106	3045	+	8	2	555			
12110	3045	AL				*Lecture Type*		
12116	3045	555						
12117	3045	-all						
12117	3045	+	2					
12120	3045	-all						
12121	3045	-all						
12121	3045	+	C108			*Location hostKey*		
12122	3045	-all						
12123	3045	-all						
12124	3045	-all						
12150	3045	0						
12151	3045	-all						
12152	3045	-all						
12155	3045	0						
12156	3045	-all						
12157	3045	-all						
12160	3045	1						
12161	3045	-all						
12161	3045	+	C108					
12162	3045	-all						
12165	3045	0						
12166	3045	-all						
12167	3045	-all						
12200	3045	0						
12201	3045	7200						
12202	3045	0						
12203	3045	5						
12204	3045	-all						
12205	3045	1						
12206	3045	1						
12331	3045	-all						
12331	3045	+						
12332	3045							
12333	3045							
12334	3045							

12335 3045
12401 3045
REM

This activity specification is very similar to module specification, it has the hostkey 3045, the activities are identified with Id 12000 and it has the type and in this case, it is a Lecture (AL).

Considerations of dealing with data

To process with the data file, I read each line of the file and then identify the text string to change. Thus the original file is open, read the line and make the permanent changes in a new text file.

At the beginning, to deal with data file I have taken the most important attributes of the file, these attributes are hostkey, name, description, nDepartment and Term in the module case and hostkey, name, description, nDepartment and nModule in the activity case. These data have been taken to introduce in the Access program table to assist the programs to makes changes, imported ID and hostkey data for modules and activities were imported into an Access database. This information provides information to the changes to be made

Programs

The project concentrates on a 'find and replace' exercise.

Initially I investigated with Microsoft Access as a means of data string substitution not consistent. However, as the data structure of the UEF file changed, it was not possible to import the data in a uniform format and some part would have been necessary to return the database export into the correct UEF format. Alternatively, the UEF file could have been into different tables running different data sets. So, the format of UEF file meant that it could not be possible to out all data in one table, this one did that I need multiple tables, cut part of the data and paste in original file, multiple queries, etc.

Also, the uef file could need to be construction from data outputs for various Access tables. If Syllabus Plus was sensitive to data format, problems may have solution importing a prefabricated data file.

It was decided against using Access but to use a Visual Basic program to read the UEF file. The advantaged of this method is that the modified file could have identical format to the original file and the new file created could be successfully imported into Syllabus Plus.

However, I was used Access because it could be used to organize the text file dates, even though I could use other program like Mysql, Sql, etc. to do the tables, I decided to use Access because I think that it is very intuitive and useful.

Finally I used it to do the main tables of Module and Activity and after I can do the relevant queries in the tables for then it will be used in my Visual Basic program.

One table is called Module, it has the following fields:

- HostKey: it has all modules hostkey
- Title: it has every name of all modules from file.
- Description: it has the description of every module
- nDepartment: it is the department number that belong every module
- Term: it is the term for every module, it was necessary for after to do several modification in the week pattern from term to semester.

A second table is called Activity, it has the following fields:

- HostKey: it has all activities hostkey
- Title: it has every name of all activities from file
- Description: it has the description of every activity
- nDepartment: it is the department number that belong every activity
- nModule: it is the module hostkey and it references to module that the activity belongs.
- ActivityType: it is the type from activity, it can be lecture, tutorial, lab, etc.

After done with the tables, I continued the project with visual basic and the tables contruction.

Visual Basic provides many interesting sets of tools to aid in building applications. This program can add a substantial amount of code simply by controls, such as buttons and dialog boxes and then defining their appearance and behaviour.

I decided to use VB because I thought it is the better way to read a file, to do several changes and after to create a file of exit with all changes.

As I wanted to use in VB program the tables that I had created in Access I had to configure some things in the Control Panel to connect VB and Access.

I followed the next configurations:

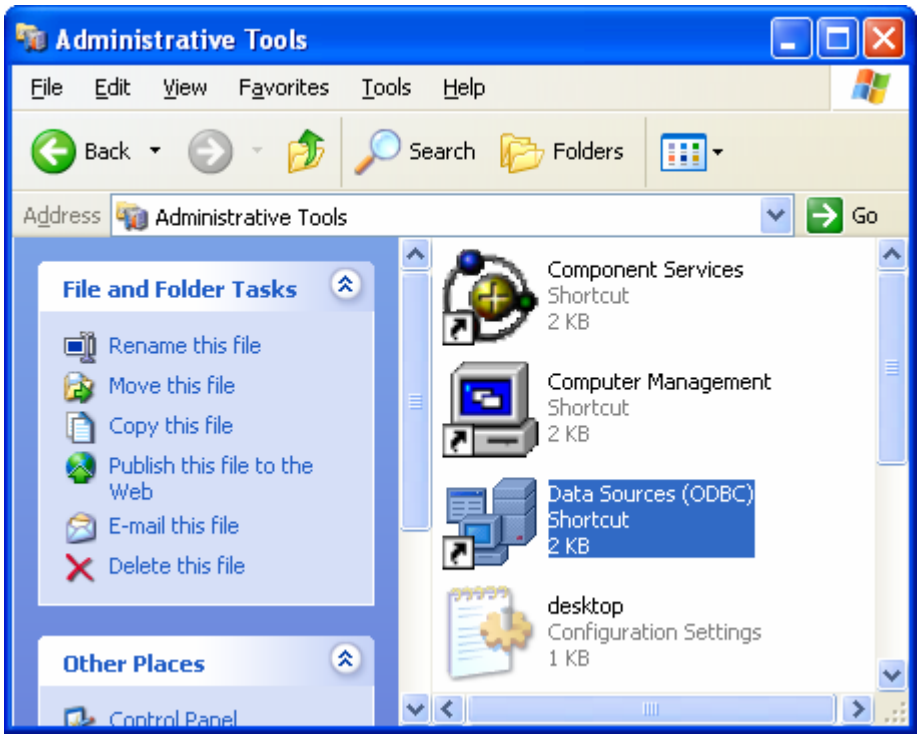


Figure 1. Control Panel > Administrative tools > Data Sources (ODBC)

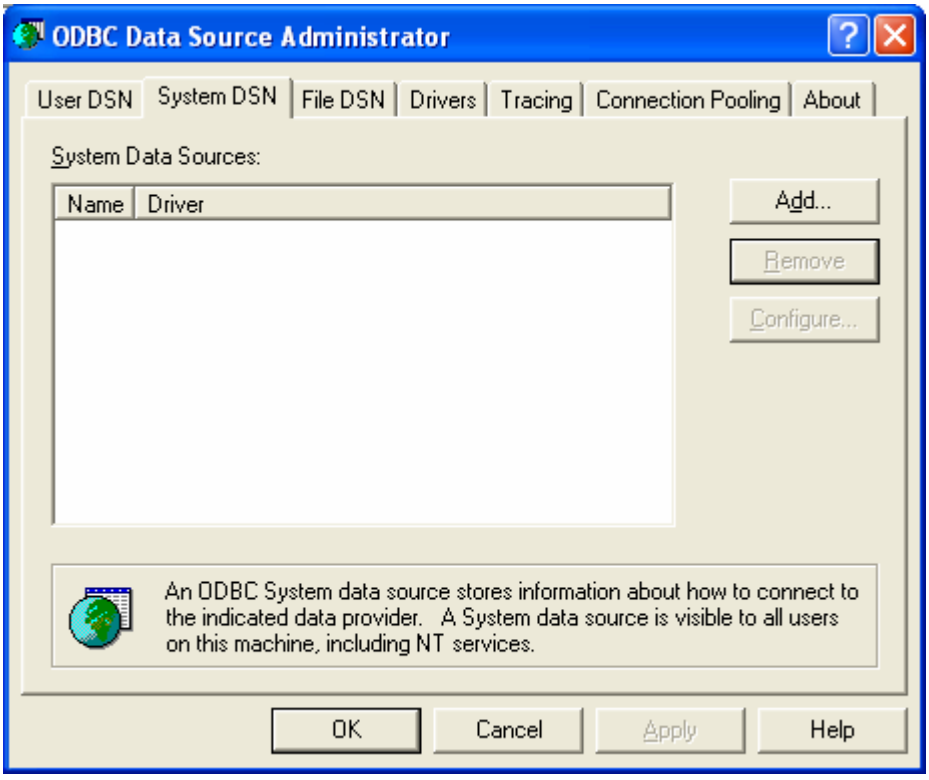


Figure 2. The ODBC Data Source Administrator: select the "System DSN" tag and click "Add"

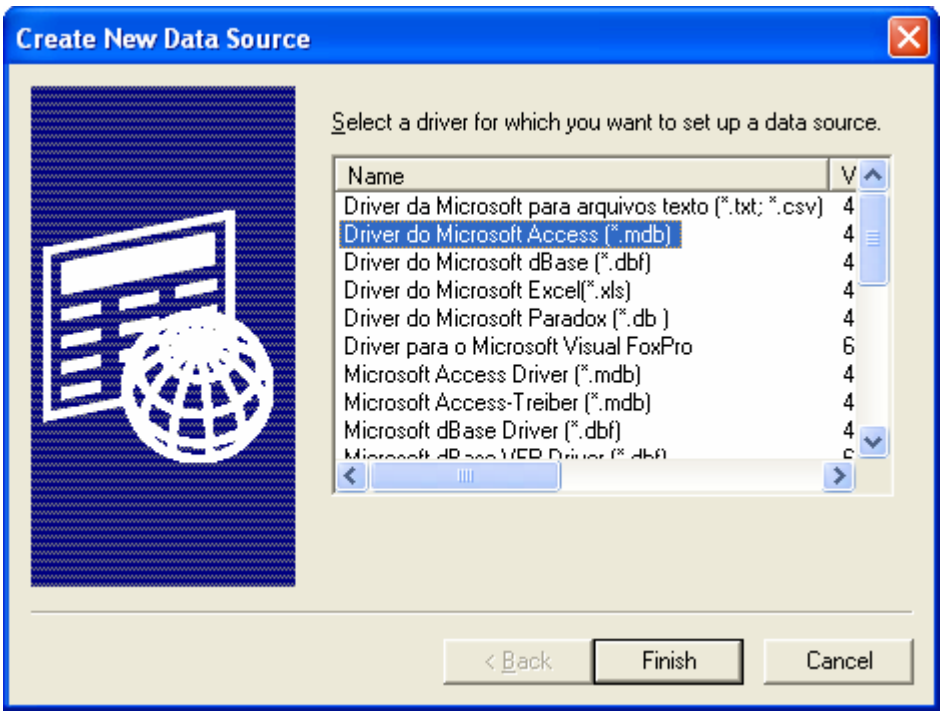


Figure 3 .Creating a New Data Source: I need to highlight "Driver do Microsoft Access (*.mdb)" and click "Finish"

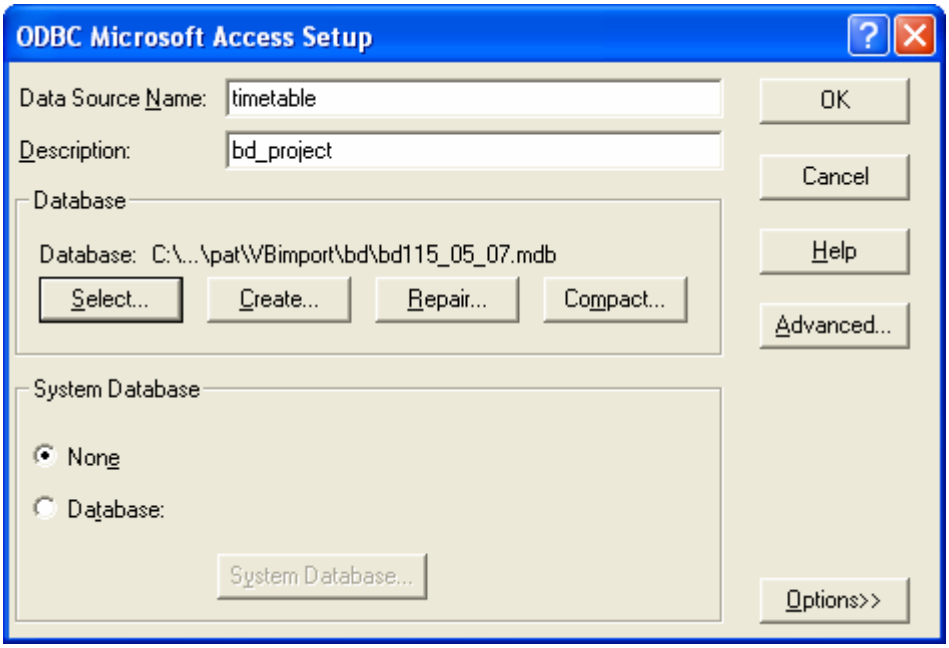


Figure 4. ODBC Microsoft Access Setup: First of all, give the Data Source a name and keep it simple. Then, give it a friendly description in case other Administrators need to know what it's for. Finally, you'll need to point ODBC at the database you created in step 1. Leave all the other options as they are and click OK.

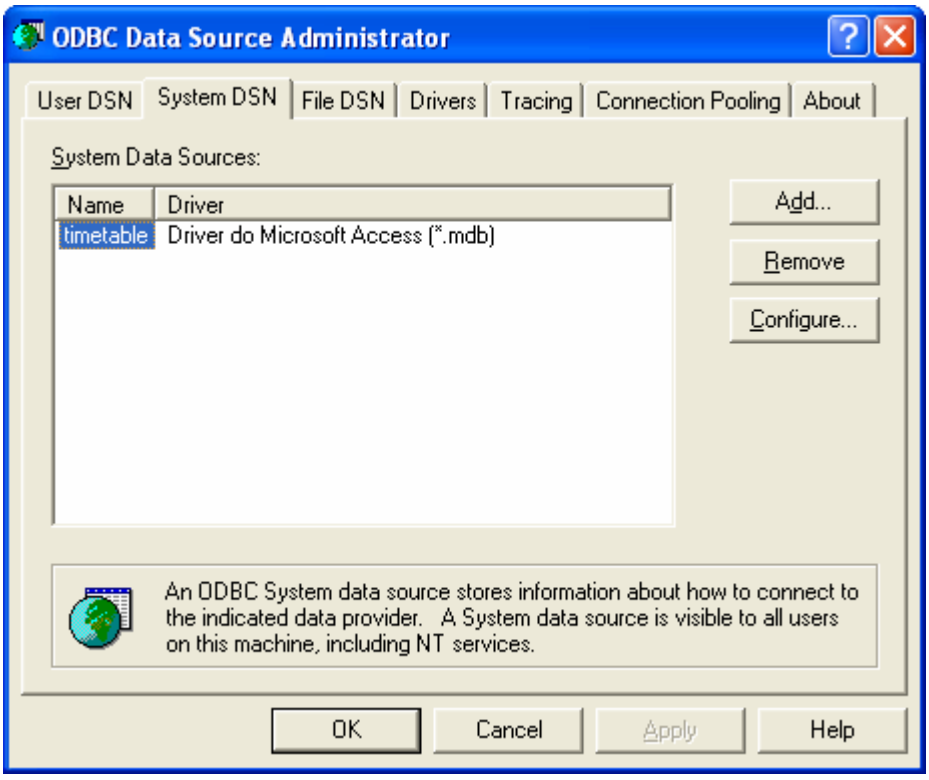


Figure 5. ODBC with new database connection: The ODBC dialogue box should look something like figure in above.

Program 1: *Import Database*

This first program is to top up the file dates in the Module table and Activity table. In the same program, I need the file date to complete every fields of the table like to activity table, the one difference that there is of one table to the other is for module table I have taken file line 11063 to complete the terms and for activity I have taken file line 12022 and 12110 to complete the module number and activity type.
To complete the rest of table fields I have taken file line 11004 to complete the hostkey and name of module, 11005 to complete the description and 11020 the department number, activity table needs the same but in this case I have taken file line 12004, 12005 and 12020.

For adding, every date in every field, I have used an Ado

ActiveX Data objects (ado)

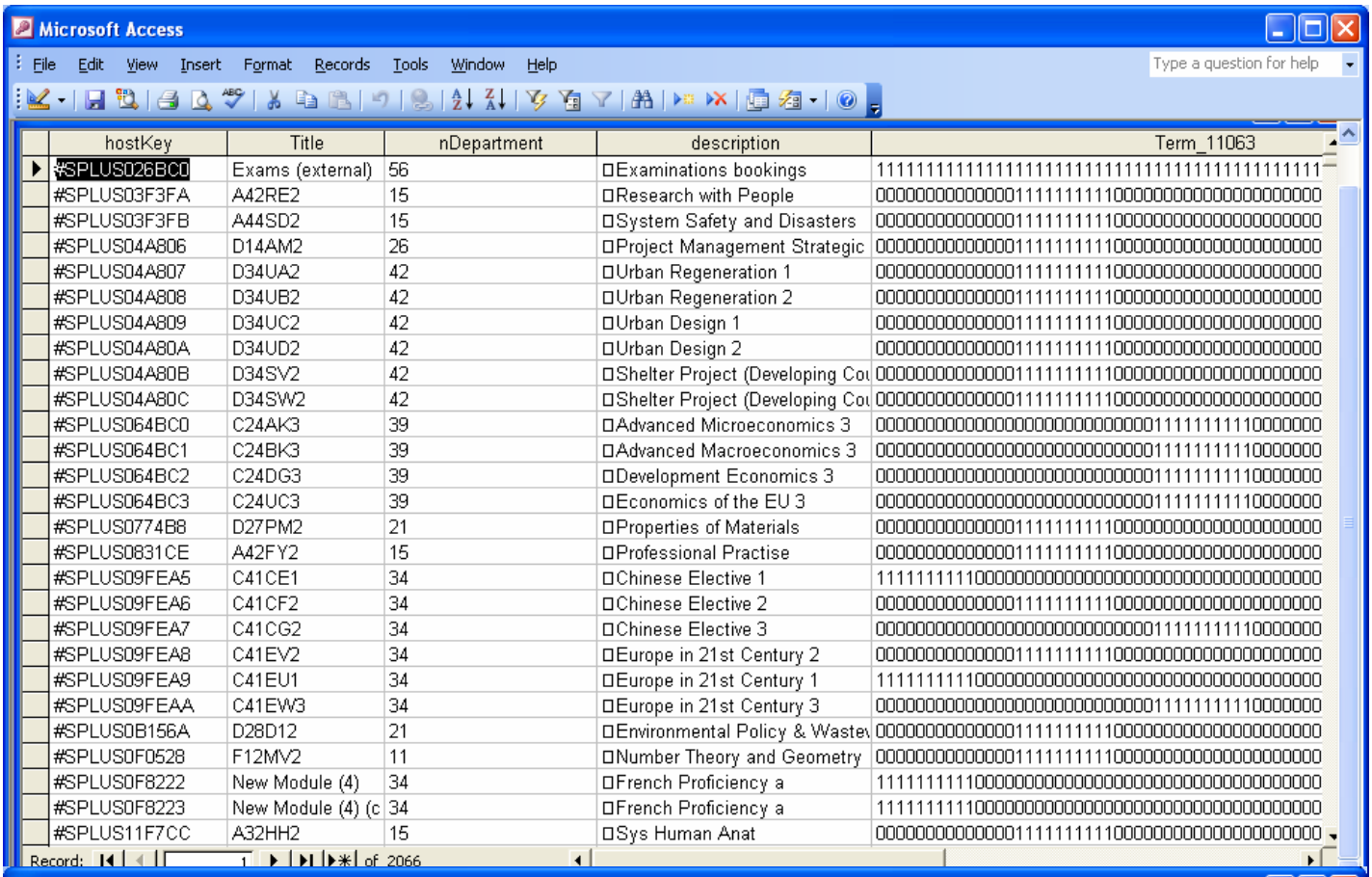
There defines a model of programming which provides the necessary elements to have access and to update a database.

Ado provides the mechanisms to realize the following activities:

1. To connect to a net
2. To specify a command to have access to the database
3. To execute a command.
4. To store the rows of a table as response of the execution of command.
5. Update the database
6. To provide a mechanism to detect the mistakes

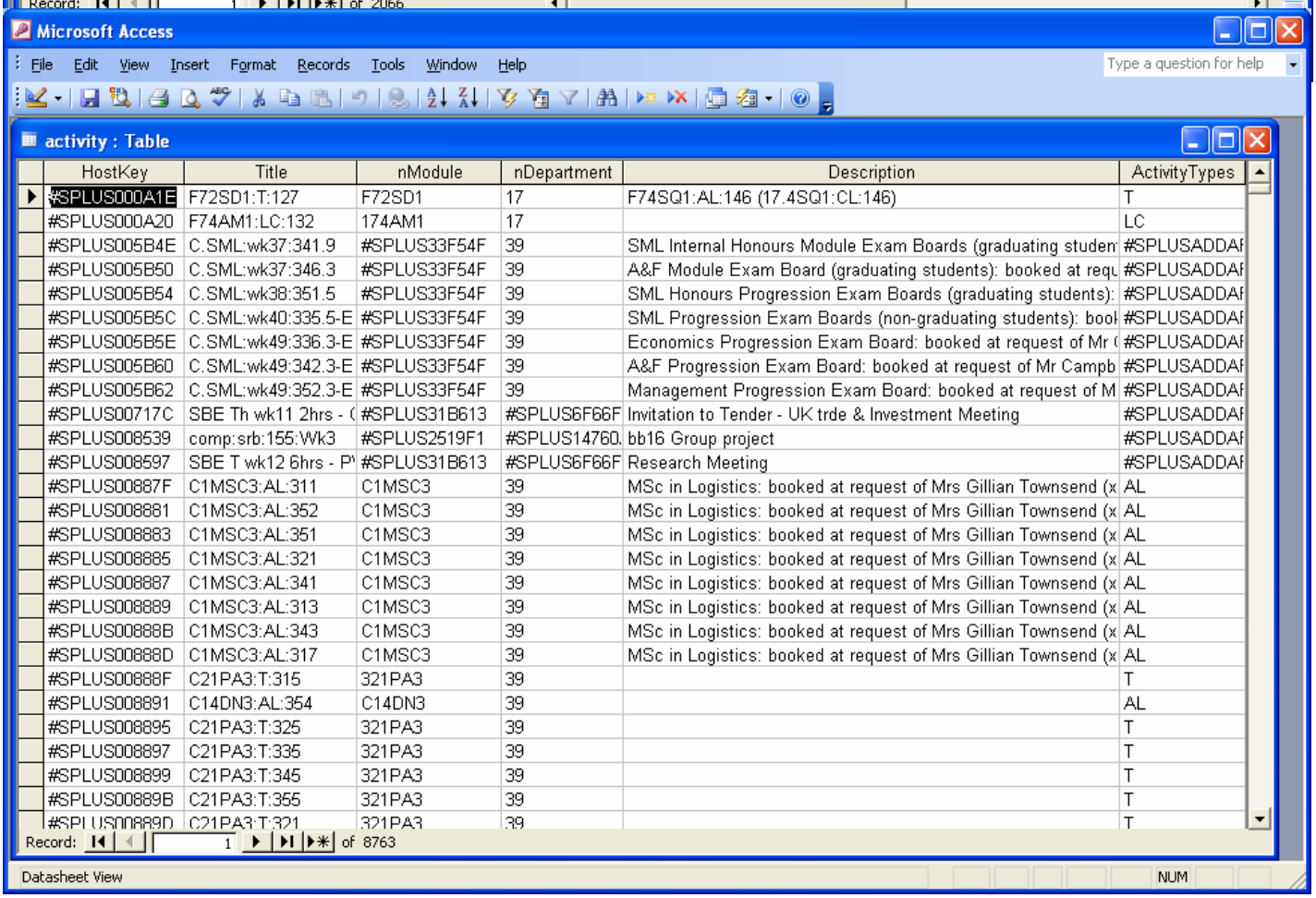
With the Ado called Adodc1, I am going to add the dates in the respective fields

Adodc1 has the next select: “select * from activity” and then with the command Adodc1.AddNew add in every fields the date file that I found in the file lines.
With module is the same code but I need to add the case when the line is 11063 to complete the terms of every module, in the module program I don’t need number of module and activity type.
I use a function called *nCaractersUntilEmpty* to move in the differences position of the line, this function return the number of characters before to find a space, so I can catch the date to then complete the fields.
The function “Mid” is particularly from Visual Basic, this function return the date from one length to other length.
The program read all lines date and when found the lines put every thing in table fields.



hostKey	Title	nDepartment	description	Term_11063
#SPLUS026BC0	Exams (external)	56	Examinations bookings	11111111111111111111111111111111111111111111111
#SPLUS03F3FA	A42RE2	15	Research with People	00000000000000011111111110000000000000000000000
#SPLUS03F3FB	A44SD2	15	System Safety and Disasters	00000000000000011111111110000000000000000000000
#SPLUS04A806	D14AM2	26	Project Management Strategic	00000000000000011111111110000000000000000000000
#SPLUS04A807	D34UA2	42	Urban Regeneration 1	00000000000000011111111110000000000000000000000
#SPLUS04A808	D34UB2	42	Urban Regeneration 2	00000000000000011111111110000000000000000000000
#SPLUS04A809	D34UC2	42	Urban Design 1	00000000000000011111111110000000000000000000000
#SPLUS04A80A	D34UD2	42	Urban Design 2	00000000000000011111111110000000000000000000000
#SPLUS04A80B	D34SV2	42	Shelter Project (Developing Cou	00000000000000011111111110000000000000000000000
#SPLUS04A80C	D34SW2	42	Shelter Project (Developing Cou	00000000000000011111111110000000000000000000000
#SPLUS064BC0	C24AK3	39	Advanced Microeconomics 3	00000000000000000000000000011111111110000000
#SPLUS064BC1	C24BK3	39	Advanced Macroeconomics 3	00000000000000000000000000011111111110000000
#SPLUS064BC2	C24DG3	39	Development Economics 3	00000000000000000000000000011111111110000000
#SPLUS064BC3	C24UC3	39	Economics of the EU 3	00000000000000000000000000011111111110000000
#SPLUS0774B8	D27PM2	21	Properties of Materials	00000000000000011111111110000000000000000000000
#SPLUS0831CE	A42FY2	15	Professional Practise	00000000000000011111111110000000000000000000000
#SPLUS09FEA5	C41CE1	34	Chinese Elective 1	11111111110000000000000000000000000000000000000
#SPLUS09FEA6	C41CF2	34	Chinese Elective 2	00000000000000011111111110000000000000000000000
#SPLUS09FEA7	C41CG2	34	Chinese Elective 3	00000000000000000000000000011111111110000000
#SPLUS09FEA8	C41EV2	34	Europe in 21st Century 2	00000000000000011111111110000000000000000000000
#SPLUS09FEA9	C41EU1	34	Europe in 21st Century 1	11111111110000000000000000000000000000000000000
#SPLUS09FEAA	C41EW3	34	Europe in 21st Century 3	00000000000000000000000000011111111110000000
#SPLUS0B156A	D28D12	21	Environmental Policy & Wastev	00000000000000011111111110000000000000000000000
#SPLUS0F0528	F12MV2	11	Number Theory and Geometry	00000000000000011111111110000000000000000000000
#SPLUS0F8222	New Module (4)	34	French Proficiency a	11111111110000000000000000000000000000000000000
#SPLUS0F8223	New Module (4) (c	34	French Proficiency a	11111111110000000000000000000000000000000000000
#SPLUS11F7CC	A32HH2	15	Sys Human Anat	00000000000000011111111110000000000000000000000

Record: 1 of 2066



HostKey	Title	nModule	nDepartment	Description	ActivityTypes
#SPLUS000A1E	F72SD1:T:127	F72SD1	17	F74SQ1:AL:146 (17.4SQ1:CL:146)	T
#SPLUS000A20	F74AM1:LC:132	174AM1	17		LC
#SPLUS005B4E	C.SML:wk37:341.9	#SPLUS33F54F	39	SML Internal Honours Module Exam Boards (graduating studen	#SPLUSADDAF
#SPLUS005B50	C.SML:wk37:346.3	#SPLUS33F54F	39	A&F Module Exam Board (graduating students): booked at requ	#SPLUSADDAF
#SPLUS005B54	C.SML:wk38:351.5	#SPLUS33F54F	39	SML Honours Progression Exam Boards (graduating students):	#SPLUSADDAF
#SPLUS005B5C	C.SML:wk40:335.5-E	#SPLUS33F54F	39	SML Progression Exam Boards (non-graduating students): bool	#SPLUSADDAF
#SPLUS005B5E	C.SML:wk49:336.3-E	#SPLUS33F54F	39	Economics Progression Exam Board: booked at request of Mr (	#SPLUSADDAF
#SPLUS005B60	C.SML:wk49:342.3-E	#SPLUS33F54F	39	A&F Progression Exam Board: booked at request of Mr Campb	#SPLUSADDAF
#SPLUS005B62	C.SML:wk49:352.3-E	#SPLUS33F54F	39	Management Progression Exam Board: booked at request of M	#SPLUSADDAF
#SPLUS00717C	SBE Th wk11 2hrs - (	#SPLUS31B613	#SPLUS6F66F	Invitation to Tender - UK trde & Investment Meeting	#SPLUSADDAF
#SPLUS008539	comp:srb:155:Wk3	#SPLUS2519F1	#SPLUS14760.	bb16 Group project	#SPLUSADDAF
#SPLUS008597	SBE T wk12 6hrs - P'	#SPLUS31B613	#SPLUS6F66F	Research Meeting	#SPLUSADDAF
#SPLUS00887F	C1MSC3:AL:311	C1MSC3	39	MSc in Logistics: booked at request of Mrs Gillian Townsend (x	AL
#SPLUS008881	C1MSC3:AL:352	C1MSC3	39	MSc in Logistics: booked at request of Mrs Gillian Townsend (x	AL
#SPLUS008883	C1MSC3:AL:351	C1MSC3	39	MSc in Logistics: booked at request of Mrs Gillian Townsend (x	AL
#SPLUS008885	C1MSC3:AL:321	C1MSC3	39	MSc in Logistics: booked at request of Mrs Gillian Townsend (x	AL
#SPLUS008887	C1MSC3:AL:341	C1MSC3	39	MSc in Logistics: booked at request of Mrs Gillian Townsend (x	AL
#SPLUS008889	C1MSC3:AL:313	C1MSC3	39	MSc in Logistics: booked at request of Mrs Gillian Townsend (x	AL
#SPLUS00888B	C1MSC3:AL:343	C1MSC3	39	MSc in Logistics: booked at request of Mrs Gillian Townsend (x	AL
#SPLUS00888D	C1MSC3:AL:317	C1MSC3	39	MSc in Logistics: booked at request of Mrs Gillian Townsend (x	AL
#SPLUS00888F	C21PA3:T:315	321PA3	39		T
#SPLUS008891	C14DN3:AL:354	C14DN3	39		AL
#SPLUS008895	C21PA3:T:325	321PA3	39		T
#SPLUS008897	C21PA3:T:335	321PA3	39		T
#SPLUS008899	C21PA3:T:345	321PA3	39		T
#SPLUS00889B	C21PA3:T:355	321PA3	39		T
#SPLUS00889D	C21PA3:T:321	321PA3	39		T

Record: 1 of 8763

Datasheet View NUM

Program 2: *Add population activities*

Module Name	Number Population	
B31CI3	3	
B31DI2	109	
B31EI3	128	
B31EN1	105	
B31MY2	14	
B31PX1	55	
B31PY2	146	
B31PZ3	52	
B32BD1	8	
B32DO3	66	
B32EO1	143	
B32FG2	57	
B32GI1	52	
B32MH3	3	
B32PR2	116	hostkey 222PR2
B32RA1	51	
B32RP3	51	
B32SF3	58	
B32UA2	17	
B33CK2	57	

A list of selected modules and their population were used to test the program that added class sizes to modules and lecture activities. This list had module code and module population
To change the population activity I need to know the hostkey of the Module Name, so at first I found the hostkey of every module name and after I can change the population activity of that activity which has a nModule the hostkey that I have found. I only have to change the population activity in those activities which have the Activity Types like a Lecture (AL).
This program use 2 Adodc, the first one is called Adodc2 and the second one Adodc3. The Adodc2 select all hostkey from module where the title is one of that Module name.

So I have next select:

Sql= “select hostkey from module where title=’ B34VJ2’ and ‘B34VI1’….” This select is assigned to the first one (Adodc2).

The second one Adodc3, has the next select:

Sql1= “select title from activity where nModule=!Adodc2.Recordset!hostkey” and then when I found the line REM Activity: ***title*** I check if this title belong to sql1. Finally I change the number in line 12021 and 12039

```
REM Activity: B32PR2:AL:243
12004  5862   B32PR2:AL:243
12005  5862   B32PR2:AL:243
12020  5862   #SPLUS147609
12036  5862   0
12021  5862   0
12022  5862   +        222PR2        0
12028  5862   1
12031  5862   -all
12032  5862   -all
12035  5862
12039  5862   0
12051  5862   1
```

```
REM Activity: B32PR2:AL:243
12004  5862   B32PR2:AL:243
12005  5862   B32PR2:AL:243
12020  5862   #SPLUS147609
12036  5862   0
12021  5862   116
12022  5862   +        222PR2        0
12028  5862   1
12031  5862   -all
12032  5862   -all
12035  5862
12039  5862   116
12051  5862   1
```

Program 3: *Add population module*

In this program I only need one Adod to search the module name in the module table, so when I found a line as “REM Module: title” if this title is the same to some title of the Adodc, then it will be necessary to change the population number.
So I use the same adod that I used to add population activity, the other one is not necessary in this case, the select that I have created to allow at Adod is:

Sql= “select hostkey from module where title=’ B34VJ2’ and ‘B34VI1’….”

I change the population module in line 11021 and 11039 as well.

```
REM Module: B32PR2
11004  222PR2        B32PR2
11005  222PR2        Prog for Eng
11020  222PR2        #SPLUS147609
11036  222PR2        0
11021  222PR2        0
11027  222PR2        -all            0
11027  222PR2        +       22112   0
11027  222PR2        +       #SPLUS4D35A9        0
11027  222PR2        +       23112   0
11027  222PR2        +       #SPLUSF99AFB        0
11027  222PR2        +       #SPLUSF99AFF        0
11027  222PR2        +       #SPLUS5AA302        0
11027  222PR2        +       #SPLUS4D35A7        0
11027  222PR2        -all            1
11030  222PR2        -all
11039  222PR2        0
11051  222PR2        1

REM Module: B32PR2
11004  222PR2        B32PR2
11005  222PR2        Prog for Eng
11020  222PR2        #SPLUS147609
11036  222PR2        0
11021  222PR2        116
11027  222PR2        -all            0
11027  222PR2        +       22112   0
11027  222PR2        +       #SPLUS4D35A9        0
11027  222PR2        +       23112   0
11027  222PR2        +       #SPLUSF99AFB        0
11027  222PR2        +       #SPLUSF99AFF        0
11027  222PR2        +       #SPLUS5AA302        0
11027  222PR2        +       #SPLUS4D35A7        0
11027  222PR2        -all            1
11030  222PR2        -all
11039  222PR2        116
11051  222PR2        1
```

Program 4: *Delete term 3*

To delete term 3, I use one Adod, this one select the title of module who belong to term3, so I have:

sql1 = "select Title, hostKey from module where Term_11063 = '0000000000000000000000000000111111111100000000000000' OR Term_11063= '0000000000000000000000000000111111111000000000000000'"

The module which belong to term 3 have the field Term_11063 equal to:

0000000000000000000000000000111111111100000000000000 or
0000000000000000000000000000111111111000000000000000

I allow the sentences sql to Adod, so I have:

Adodc2.Recorsource=sql1

So I can delete all modules of term3, because I have the title and with it when I found one line as: REM module: *title* if the title is the same with the Adod, it will be delete.

…….

```
11331   222UA2  +
11332   222UA2
11333   222UA2
11334   222UA2
11335   222UA2
REM
REM Module: B51LB3
11004   #SPLUS2F6B12    B51LB3
11005   #SPLUS2F6B12    Eng Coursework 2
11020   #SPLUS2F6B12    #SPLUS28A305
11036   #SPLUS2F6B12    0
11021   #SPLUS2F6B12    0
11027   #SPLUS2F6B12    -all        0
11027   #SPLUS2F6B12    -all        1
11030   #SPLUS2F6B12    -all
11039   #SPLUS2F6B12    0
11051   #SPLUS2F6B12    1
11051   #SPLUS2F6B12    0
11061   #SPLUS2F6B12    196     264
11063   #SPLUS2F6B12    0000000000000000000000000000111111111100000000000000
11081   #SPLUS2F6B12    -all    0000000000000000000000000000000000000000000000000000    0   0   0
11081   #SPLUS2F6B12    +       1111111111111111111111111111000000000011111111111111    0   0
        1095
11081   #SPLUS2F6B12    +       1111111111111111111111111111000000000011111111111111    1   0
        1095
11081   #SPLUS2F6B12    +       1111111111111111111111111111000000000011111111111111    2   0
        1095
11081   #SPLUS2F6B12    +       1111111111111111111111111111000000000011111111111111    3   0
        1095
11081   #SPLUS2F6B12    +       1111111111111111111111111111000000000011111111111111    4   0
        1095
11331   #SPLUS2F6B12    -all
11331   #SPLUS2F6B12    +
11332   #SPLUS2F6B12
11333   #SPLUS2F6B12
11334   #SPLUS2F6B12
11335   #SPLUS2F6B12
REM
REM Module: A23OC2
11004   A23OC2  A23OC2
11005   A23OC2  Rand Proc & Con
11020   A23OC2  15
11036   A23OC2  0
11021   A23OC2  0
```

…..

```
11331  222UA2        +
11332  222UA2
11333  222UA2
11334  222UA2
11335  222UA2
REM
REM
REM Module: A23OC2
11004  A23OC2        LS3OC2
11005  A23OC2        Rand Proc & Con
11020  A23OC2        15
11036  A23OC2        0
11021  A23OC2        0

......
```

Program 5: *Change name of module (activities who belong to module) and change the teaching week pattern from term to semester*

This is really a complex program, because on the one hand change the name of modules and at the same time it changes the activity name which belongs to the module that I have changed the name before.
On the second hand, this program changes the module terms to semester.

I have created a new table in access with the all old modules names and the new modules names, so I have searched the old code name in the file and to compare with the module name of the file, if this are equals then I change it.

An example of some old codes and new codes that It is necessary to change is:

module_codes	
old_code	**new_code**
A11AB2	LS1AB2
A11BP2	LS1BP2
A12GP2	LS2GP2
A12LB2	LS2LB2
A12MB2	LS2MB2
A13BR2	LS3BR2
A13CP2	LS3CP2
A13MC2	LS3MC2
A13MP2	LS3MP2
A13OM2	LS3OM2
A13TC2	LS3TC2
A14BD2	LS4BD2
A14BE2	LS4BE2
A14BF2	LS4BF2
A14CD2	LS4CD2
A14CE2	LS4CE2
A14CF2	LS4CF2
A14FC2	LS4FC2

To do these changes I have created a sql sentences to then allows to Adod:

```
sql = "select old_code, new_code from module_codes"
Adodc2.RecordSource = sql
```

After that, I am going to read the file and when I found the line 11004 I check if the name is in the Adod, line 11004 is where I found the module name, so I have:

If myLine = Adodc2.Recordset!old_code then I change the name with the new code name.

I do the same with I found line 12004, this line is the activity name, if the part of activity name is equal to old_code I change, but the activity name always is longer than the old code, so I compare the first 6 digits of the activity name, so I have:

If (Mid(myLine, 1, 6) = Adodc2.Recordset!old_code) then I change the activity name.

myLine always has the name of module or activity this one I found in line 11004 in the case of module and line 12004 in the case of activity.

REM Module: A12LB2
11004 152GL2 A12LB2

REM Module: A12LB2
11004 152GL2 LS2LB2

REM Activity: A12LB2:AL:214
12004 486 A12LB2:AL:214

REM Activity: A12LB2:AL:214
12004 486 LS2LB2:AL:214

Terms to semester

Before explaining how I have changed the terms to semester, I would like to explain how the terms are structured.

I found the terms in line 11063 in the module case and 12063 in the activity case, so I found for example:

Module case (term 1):

11063 344SI1 11111111110000000000000000000000000000000000000000

Activity case (term 1):

12063 3045 11111111100000000000000000000000000000000000000000

In these two cases the module and activity belong to term 1, but we can find modules and activities with term 2 or 3 like:

Term 2:

11063 A33RE2 00000000000000111111111100000000000000000000000000

12063 #SPLUSFFBA44 00000000000000111111100000000000000000000000000000

Term 3:

11063 D34SS3 00000000000000000000000000011111111110000000000000

12063 #SPLUSB83A35 00000000000000000000000000011111110000000000000000

Week pattern term line has 52 digits, each digits represent weeks and is structured as follows:

10 weeks(Term1)	*4 weeks(holidays)*	*10 weeks(Term2)*	*4 weeks(holiday)*	*10 Weeks(Term3)*	*14 Weeks(holidays)*
1111111111	0000	0000000000	0000	0000000000	00000000000000
0000000000	0000	1111111000	0000	0000000000	00000000000000
0000000000	0000	0000000000	0000	1111111111	00000000000000

Can be that some term doesn't complete all ten weeks, but it belong at the term in similar way.
Now I need to change the terms to semester, in semester doesn't exist term 3, like this way:

14 weeks(Seme.1)	*3 weeks(holidays)*	*12 weeks(Sem2)*	*3 weeks(holiday)*	*4Weeks(Sem. 2)*	*16 Weeks(holidays)*
11111111111111	000	000000000000	000	0000	0000000000000000
00000000000000	000	111111111111	000	1111	0000000000000000

I have two functions to know if the term is term 1 or term 2:

Shown code functions in Appendix

The first one checks, if the string belongs to term1 and the second one checks if it belongs to term2. To do this function, at first in the function to check is term 1, I have used the function Mid, the string has 52 digits, to see if it is term1 I have taken from where term 1 finish, this is in position eleven to the string finish and then if all this digits are equal to zero this string belongs to term1,
To term 2, I have taken the first fourteen digits and the last twenty eight digits and if it is equal to zero this string belongs to term 2.

To change the terms at first I 'm reading every line of file when I find the line 11063 I check with the functions isTerm1 and isTerm1 and if is Term1 I change to semester1, this one is always the same: Semester1 : 11111111111111000000000000000000000000000000000000000000 and if the line is term 2 I change with Semester 2: 00000000000000000011111111111100011110000000000000000000, is always the same as well.
After to change line 11063, I must change line 11081, this line is always opposite to line 11063, so to change line 11081 I have created one function called opposite, this function takes the line 11063 and make the opposite.

I must add that the module has a line 11081 as this, but this one is not necessary to change.

```
11081   344SI1  -all    000000000000000000000000000000000000000000000000000000000000    0       0       0
```

Show code to change the terms and function opposite in appendix

```
REM Module: D29CF2
11004   #SPLUS681C5E    D29CF2
11005   #SPLUS681C5E    Coastal Flood Defence
11020   #SPLUS681C5E    #SPLUS6F66F5
11036   #SPLUS681C5E    0
11021   #SPLUS681C5E    0
11027   #SPLUS681C5E    -all                0
11027   #SPLUS681C5E    +       #SPLUS65BC49    0
11027   #SPLUS681C5E    -all                1
11030   #SPLUS681C5E    -all
11039   #SPLUS681C5E    0
11051   #SPLUS681C5E    1
11051   #SPLUS681C5E    0
11061   #SPLUS681C5E    98      166
11063   #SPLUS681C5E    00000000000000111111111100000000000000000000000000000000
11081   #SPLUS681C5E    -all    000000000000000000000000000000000000000000000000000000000000    0       0       0
11081   #SPLUS681C5E    +       111111111111110000000000111111111111111111111111111111    0       0
        1095
11081   #SPLUS681C5E    +       111111111111110000000000111111111111111111111111111111    1       0
        1095
11081   #SPLUS681C5E    +       111111111111110000000000111111111111111111111111111111    2       0
        1095
11081   #SPLUS681C5E    +       111111111111110000000000111111111111111111111111111111    3       0
        1095
11081   #SPLUS681C5E    +       111111111111110000000000111111111111111111111111111111    4       0
        1095
11331   #SPLUS681C5E    -all
11331   #SPLUS681C5E    +
11332   #SPLUS681C5E
11333   #SPLUS681C5E
11334   #SPLUS681C5E
11335   #SPLUS681C5E
REM
```

```
REM Module: D29CF2
11004   #SPLUS681C5E   D29CF2
11005   #SPLUS681C5E   Coastal Flood Defence
11020   #SPLUS681C5E   #SPLUS6F66F5
11036   #SPLUS681C5E   0
11021   #SPLUS681C5E   0
11027   #SPLUS681C5E   -all                0
11027   #SPLUS681C5E   +          #SPLUS65BC49    0
11027   #SPLUS681C5E   -all                1
11030   #SPLUS681C5E   -all
11039   #SPLUS681C5E   0
11051   #SPLUS681C5E   1
11051   #SPLUS681C5E   0
11061   #SPLUS681C5E   98         166
11063   #SPLUS681C5E   00000000000000000111111111110001111000000000000000
11081   #SPLUS681C5E   -all        000000000000000000000000000000000000000000000000000000000000000000000000    0    0    0
11081   #SPLUS681C5E   +          11111111111111111000000000000111000011111111111111111    0    0
        1095
11081   #SPLUS681C5E   +          11111111111111111000000000000111000011111111111111111    1    0
        1095
11081   #SPLUS681C5E   +          11111111111111111000000000000111000011111111111111111    2    0
        1095
11081   #SPLUS681C5E   +          11111111111111111000000000000111000011111111111111111    3    0
        1095
11081   #SPLUS681C5E   +          11111111111111111000000000000111000011111111111111111    4    0
        1095
11331   #SPLUS681C5E   -all
11331   #SPLUS681C5E   +
11332   #SPLUS681C5E
11333   #SPLUS681C5E
11334   #SPLUS681C5E
11335   #SPLUS681C5E
REM
```

Program 6: *Change the teaching week pattern from term to semester of activity*

This program is similar to previous because it uses the same functions, isTerm1 and isTerm2. The only difference is in the line, where I have to change, this program change the term, when I find the line with number 12063 and after change line 12081 with the function called “opposite” but only these activities who have activity type as a lecture (AL)

REM Activity: C3MSC2:AL:213

12004	1326	C3MSC2:AL:213				
12005	1326					
12020	1326	39				
12036	1326	0				
12021	1326	0				
12022	1326	+	C3MSC2 0			
12028	1326	1				
12031	1326	-all				
12032	1326	-all				
12035	1326					
12039	1326	0				
12051	1326	1				
12051	1326	0				
12060	1326	60				
12061	1326	98	159			
12062	1326	9				
12063	1326	0000000000000011111111100000000000000000000000000000				
12081	1326	-all	0000000000000000000000000000000000000000000000000000	0	0	0
12081	1326	+	1111111111111100000000011111111111111111111111111111	0	0	1095
12081	1326	+	1111111111111100000000011111111111111111111111111111	1	0	1095
12081	1326	+	1111111111111100000000011111111111111111111111111111	2	0	1095
12081	1326	+	1111111111111100000000011111111111111111111111111111	3	0	1095
12081	1326	+	1111111111111100000000011111111111111111111111111111	4	0	1095
12100	1326	2				
12101	1326	0	675			
12102	1326	98	675			
12106	1326	-all				
12106	1326	+	14	0	675	
12106	1326	+	15	0	675	
12106	1326	+	16	0	675	
12106	1326	+	17	0	675	
12106	1326	+	18	0	675	
12106	1326	+	19	0	675	
12106	1326	+	20	0	675	
12106	1326	+	21	0	675	
12106	1326	+	22	0	675	
12110	1326	AL				
12116	1326	675				
12117	1326	-all				
12117	1326	+	0			
12120	1326	-all				
12121	1326	-all				
12121	1326	+	BUG20			
12122	1326	-all				
12123	1326	-all				
12124	1326	-all				
12150	1326	0				
12151	1326	-all				
12152	1326	-all				
12155	1326	0				
12156	1326	-all				
12157	1326	-all				
12160	1326	1				
12161	1326	-all				
12161	1326	+	BUG20			
12162	1326	-all				
12165	1326	0				
12166	1326	-all				
12167	1326	-all				

REM Activity: C3MSC2:AL:213

```
12004   1326    AFMSC2:AL:213
12005   1326
12020   1326    39
12036   1326    0
12021   1326    0
12022   1326    +       C3MSC2 0
12028   1326    1
12031   1326    -all
12032   1326    -all
12035   1326
12039   1326    0
12051   1326    1
12051   1326    0
12060   1326    60
12061   1326    98      159
12062   1326    9
12063   1326    00000000000000000111111111110001111000000000000000
12081   1326    -all    00000000000000000000000000000000000000000000000000    0       0       0
12081   1326    +       11111111111111111000000000000111000011111111111111    0       0       1095
12081   1326    +       11111111111111111000000000000111000011111111111111    1       0       1095
12081   1326    +       11111111111111111000000000000111000011111111111111    2       0       1095
12081   1326    +       11111111111111111000000000000111000011111111111111    3       0       1095
12081   1326    +       11111111111111111000000000000111000011111111111111    4       0       1095
12100   1326    2
12101   1326    0       675
12102   1326    98      675
12106   1326    -all
12106   1326    +       14      0       675
12106   1326    +       15      0       675
12106   1326    +       16      0       675
12106   1326    +       17      0       675
12106   1326    +       18      0       675
12106   1326    +       19      0       675
12106   1326    +       20      0       675
12106   1326    +       21      0       675
12106   1326    +       22      0       675
12110   1326    AL
12116   1326    675
12117   1326    -all
12117   1326    +       0
12120   1326    -all
12121   1326    -all
12121   1326    +       BUG20
12122   1326    -all
12123   1326    -all
12124   1326    -all
12150   1326    0
12151   1326    -all
12152   1326    -all
12155   1326    0
12156   1326    -all
12157   1326    -all
12160   1326    1
12161   1326    -all
12161   1326    +       BUG20
12162   1326    -all
12165   1326    0
12166   1326    -all
12167   1326    -
```

Results and Conclusions

The uef file created by the Visual Basic programmes were imported into Syllabus Plus an the effect on the timetable data was investigated. On each occasion it was noted that a number of issues had arisen. For example, the number of unscheduled activities appeared to increase with the addition of module population and activity population. This is an understandable result as the timetable database in its original form did not have any population data and therefore there were activities in rooms where the room may have been too small. If this event is done manually then Syllabus Plus will alert the user of this problem and will not allow the activity to be scheduled unless the scheduling constraints are set to allow this. If an uef file is uploaded into a blank database then no scheduling constraints are set. This may result in an automatic unscheduling. When the activity population was entered, 7 activities were lost from 8763 original activities. It was not possible to find these missing activities. A data file comparison was made using Access and comparing a data file extract from the original file and the imported file however each file had the same number of entries and the same number of activities without a location (unscheduled). Further investigation is needed to find a means of identifying unscheduled activities. It is important in all data conversion that no data is corrupted or lost.

When the module population was added there was no change to the number of unscheduled activities. This is expected as the module population is not used by Syllabus Plus unless activity templates are generated (these permit automatic scheduling).

When the dataset for a data file which had term 3 modules deleted and module and activity teaching patterns changed from terms to semesters was loaded into Syllabus Plus a large number of problems resulted. The following results occurred:

- Not all term 3 modules were removed
- Not all term activities were removed
- Semester Patterns were not accepted and this resulted in an increase in unavailability problems.

Some modules in the original data file did not have the typical term 3 pattern but had availability for 52 weeks. These modules and any associated activities remained. It had been assumed that if a module was removed, any activity associated with this module would also be removed (when a module is deleted manually from Syllabus Plus, activities are automatically deleted). What actually occurred was that activities not associated with a module were ‘linked’ with remaining term 3 modules rather than being deleted. The timetable software tried to keep activities rather than ignoring them.

One other major problem encountered was that the availability of modules and activities in semester 1 and semester 2 were shown as only for week 1 for semester (rather than weeks 1 to 14) or week 17 for semester 2 (rather than weeks 17 to 38??). It is not known whether or not these problems were associated with the data import form the uef file or from actions taken by Syllabus Plus when completing its import and checking. The number of problems relating to unavailability rose to over 4000 problems and these were mostly as a result of (term)semester 2 activities. The problems appeared to be ‘technical’ as when each activity was unscheduled and then rescheduled, it returned to its room and had no problems. The term 2 problems stemmed from the fact that the room availability was weeks 1, 7, 13, 19, 25, 30, 36, 41, 47, 52 rather than weeks 1 to 14 and 17 to 36.

The problems associated with the week patterns require further investigation as it was not possible to determine if the problem arose from the data set, the order in which data was changed or Syllabus Plus making decisions and amending the imported data accordingly. With further time the following tests could be done: use the Syllabus Plus programme to remove all term 3 activities (this is done by unscheduling all activities in rooms in term 3 and then deleting all unscheduled activities. Then the Visual basic programme can remove term 3 modules without ‘leaving lost term 3 actiities’. The rooms can be made available for all weeks and then amended to the semester pattern once all activities are safely imported.

Further work

If I had had more time to do this report, I would try to improve some program codes, because the program to change the activities and modules number population is late very much so I would research other way to run the program more quickly. This project, find and replace codes so if it would be use a lot, I try to do some program to change the file in interactive mode.

It is obvious that further investigation is required to ascertain whether or not Syllabus Plus is causing problems when interpreting a uef file which contains unassociated activities.

Also, it would be a worthwhile exercise converting activities into activity templates to allow future scheduling possibilities.

I

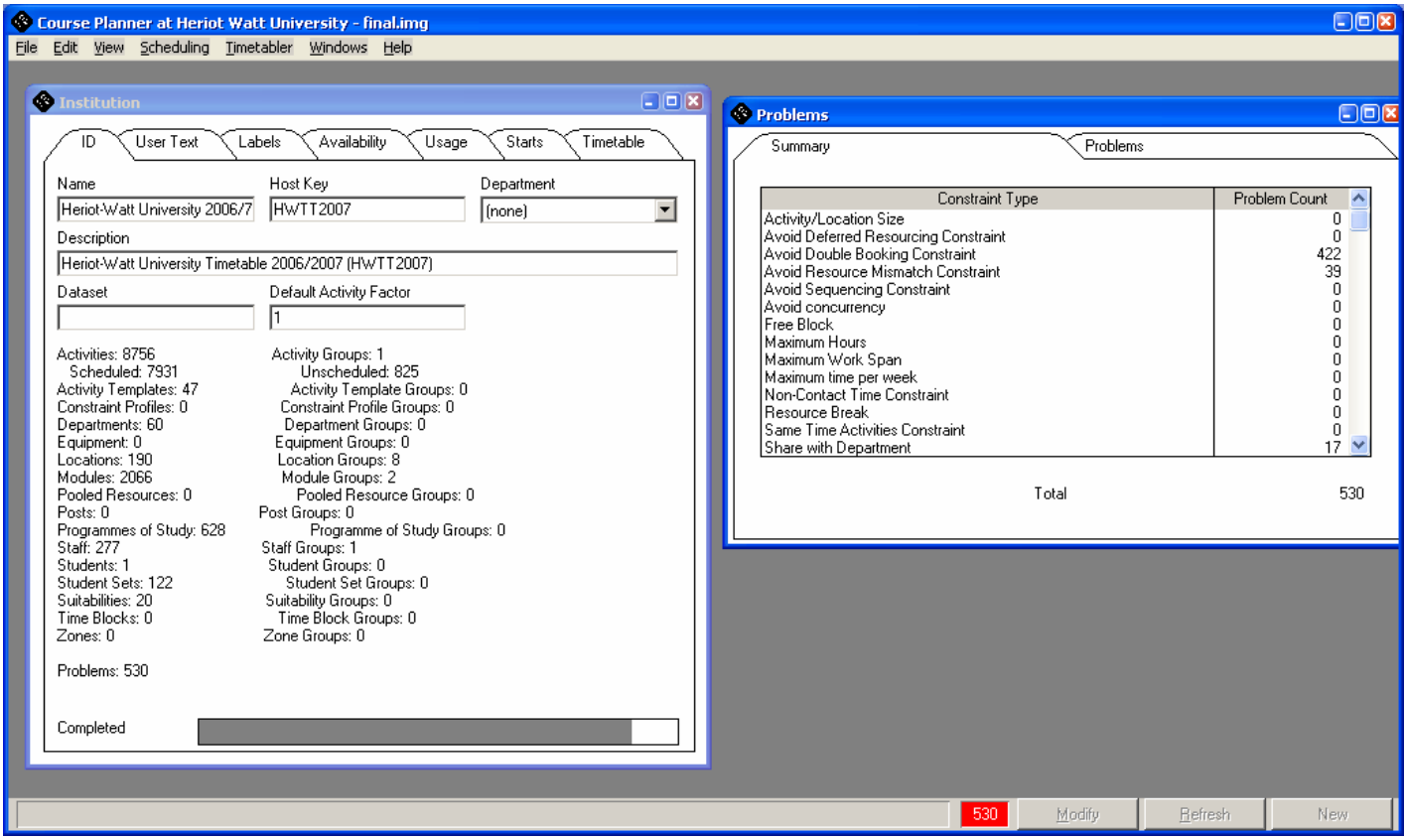


Figure1. File trial

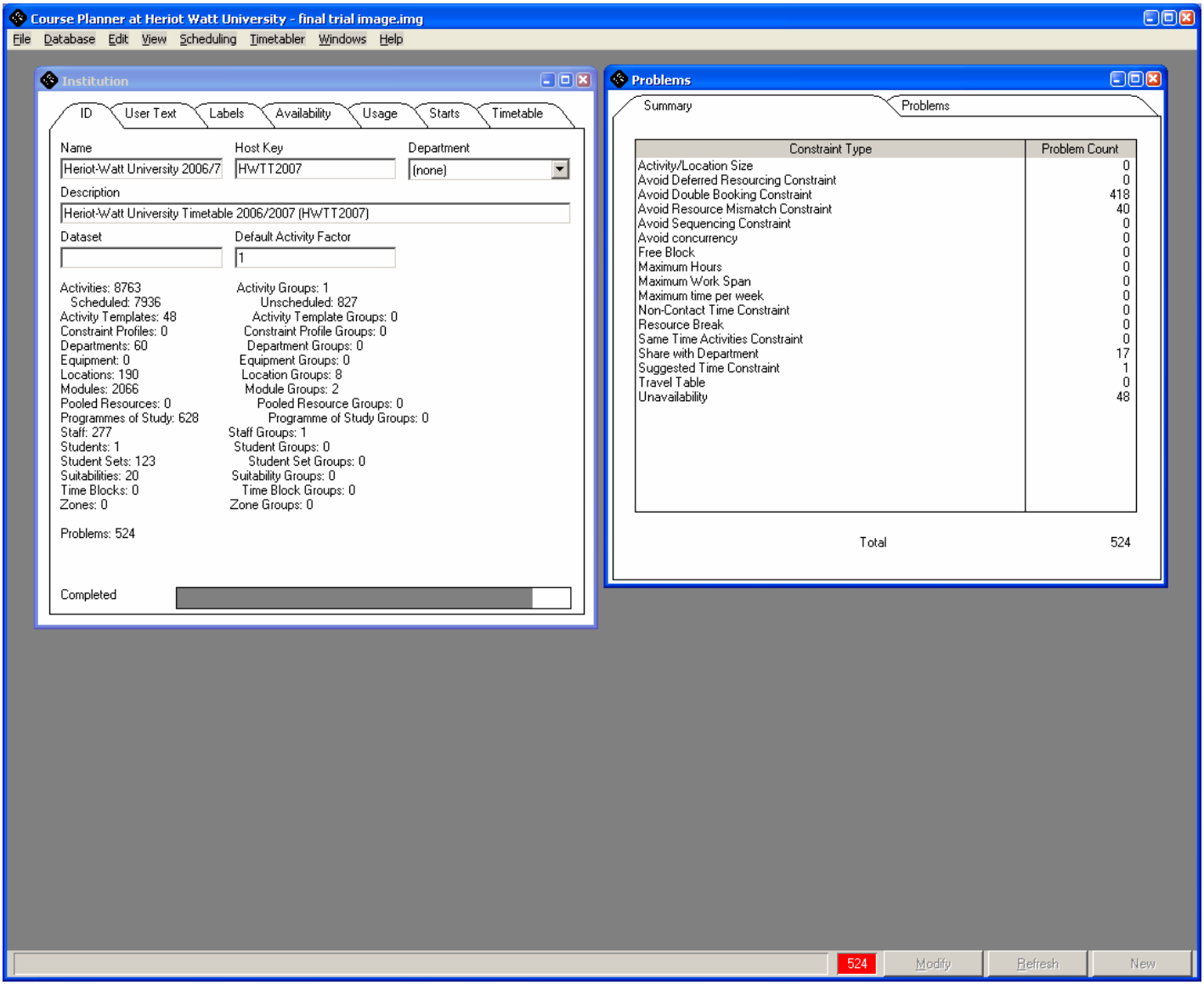


Figure2. Final file trial

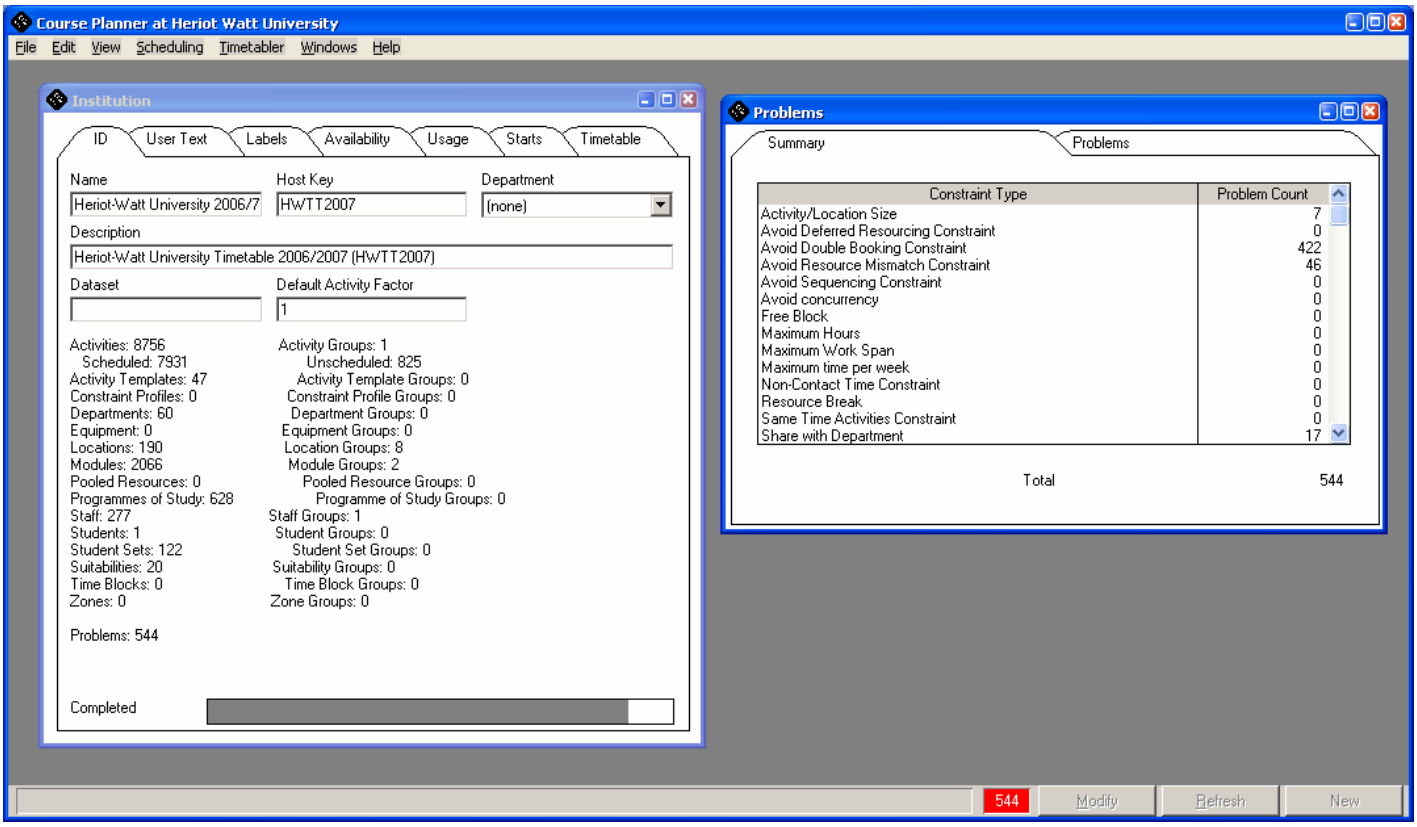


Figure 3. Change of activity population for lectures.
Increase in activity/location related problems
Double bookings
Resource mismatch

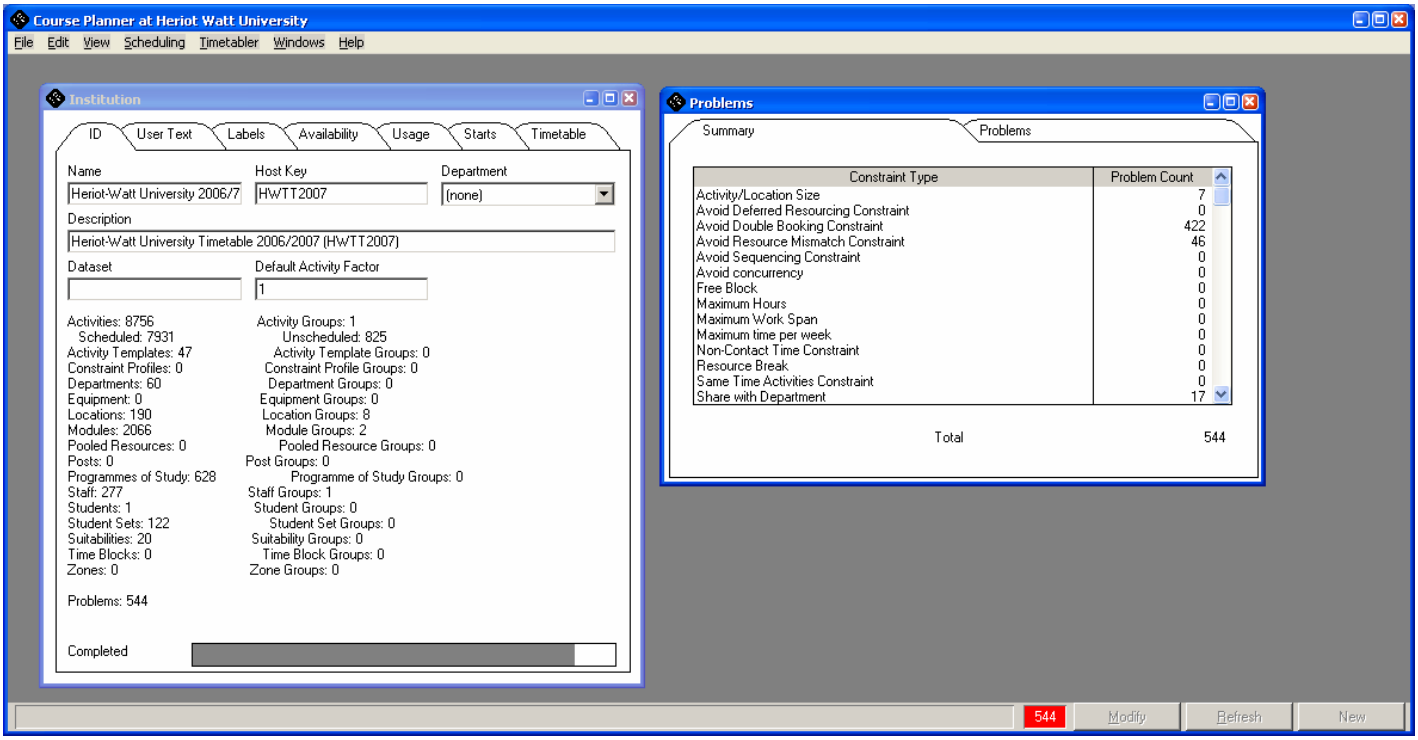


Figure 4. Change module population

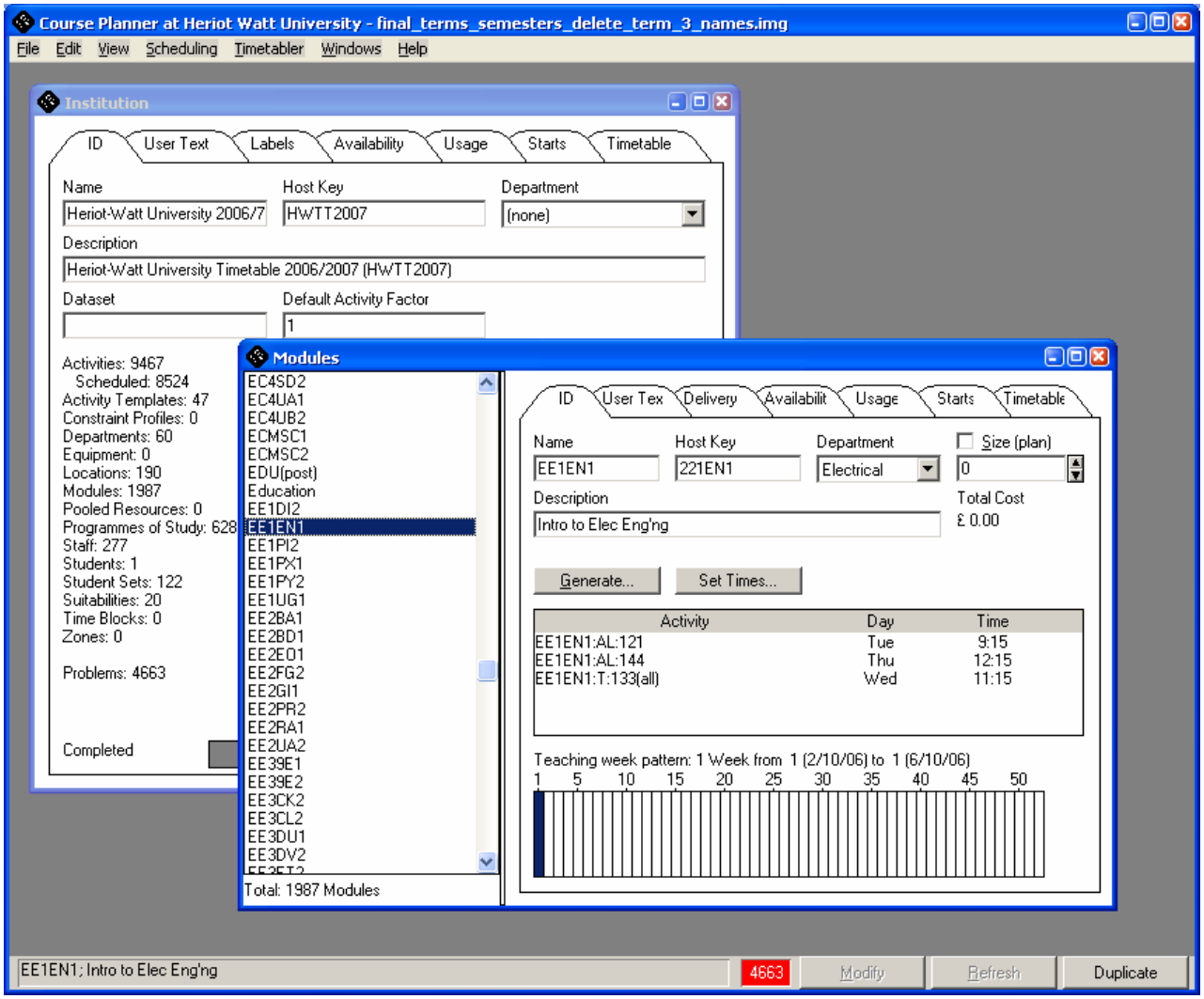


Figure 5. Screen shots showing the results from the term to weeks (delete term 3) with locations having 52 week availability (is to try and sort out the issues with non availability). The unavailability problem still exists.

Appendix

Import DataBase

```
Private Sub Command1_Click()
'-------------------------------------------------------------------------------------------------------------------------
'this program is to top up the dates file of module to Access table (use function nCaractersUntilEmpty)
'-------------------------------------------------------------------------------------------------------------------------

Dim VarTexto As String
Dim miLinia As String
Dim i As Integer
Dim j As Integer

Open "D:\pat\timetable\final_test.txt" For Input As #1

Adodc1.Refresh
Do Until EOF(1)
Line Input #1, VarTexto
i = nCaractersUntilEmpty(VarTexto, 1)
miLinia = Mid(VarTexto, 1, i) 'Take Id

VarTexto = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id

Select Case miLinia
   Case "12004"
   Adodc1.Recordset.AddNew
   Adodc1.Recordset!hostKey = Mid(VarTexto, 2, nCaractersUntilEmpty(VarTexto, i - 2) - 1)

   j = nCaractersUntilEmpty(VarTexto, 2)
   Adodc1.Recordset!title = Mid(VarTexto, j + 2, Len(VarTexto))

   Case "12005"
   j = nCaractersUntilEmpty(VarTexto, 2)
   Adodc1.Recordset!Description = Mid(VarTexto, j + 2, Len(VarTexto))

   Case "12020"
   j = nCaractersUntilEmpty(VarTexto, 2)
   Adodc1.Recordset!nDepartment = Mid(VarTexto, j + 2, Len(VarTexto))

   Case "12022"
   j = nCaractersUntilEmpty(VarTexto, 2)
   VarTexto = Mid(VarTexto, j + 1, Len(VarTexto))
   j = nCaractersUntilEmpty(VarTexto, 2)
   VarTexto = Mid(VarTexto, j + 1, Len(VarTexto))

   Adodc1.Recordset!nModule = Mid(VarTexto, j, nCaractersUntilEmpty(VarTexto, j + 1) - 1)

   Case "12110"
   j = nCaractersUntilEmpty(VarTexto, 2)
   Adodc1.Recordset!ActivityTypes = Mid(VarTexto, j + 2, Len(VarTexto))

Adodc1.Recordset.Update

End Select
Loop
Close #1
End Sub
```

```
Private Sub Command1_Click()
'-----------------------------------------------------------------------------------------------------------------------------
'this program is to top up the dates file of activity to Access table (use function nCaractersUntilEmpty)
'-----------------------------------------------------------------------------------------------------------------------------
Dim VarTexto As String
Dim miLinia As String
Dim i As Integer
Dim j As Integer

Open "D:\pat\timetable\final_test.txt" For Input As #1


Adodc1.Refresh
Do Until EOF(1)
Line Input #1, VarTexto
i = nCaractersUntilEmpty(VarTexto, 1)
miLinia = Mid(VarTexto, 1, i) 'Take Id

VarTexto = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id

Select Case miLinia
   Case "12004"
   Adodc1.Recordset.AddNew
   Adodc1.Recordset!hostKey = Mid(VarTexto, 2, nCaractersUntilEmpty(VarTexto, i - 2) - 1)

   j = nCaractersUntilEmpty(VarTexto, 2)
   Adodc1.Recordset!title = Mid(VarTexto, j + 2, Len(VarTexto))

   Case "12005"
   j = nCaractersUntilEmpty(VarTexto, 2)
   Adodc1.Recordset!Description = Mid(VarTexto, j + 2, Len(VarTexto))

   Case "12020"
   j = nCaractersUntilEmpty(VarTexto, 2)
   Adodc1.Recordset!nDepartment = Mid(VarTexto, j + 2, Len(VarTexto))

   Case "12022"
   j = nCaractersUntilEmpty(VarTexto, 2)
   VarTexto = Mid(VarTexto, j + 1, Len(VarTexto))
   j = nCaractersUntilEmpty(VarTexto, 2)
   VarTexto = Mid(VarTexto, j + 1, Len(VarTexto))

   Adodc1.Recordset!nModule = Mid(VarTexto, j, nCaractersUntilEmpty(VarTexto, j + 1) - 1)

   Case "12110"
   j = nCaractersUntilEmpty(VarTexto, 2)
   Adodc1.Recordset!ActivityTypes = Mid(VarTexto, j + 2, Len(VarTexto))

Adodc1.Recordset.Update



End Select

Loop

Close #1

End Sub
```

Change number population

```
Private Sub Command4_Click()
'-------------------------------------------------------------------------------------------------------------------
'this program is to change the number population (use function number_population_module)
'-------------------------------------------------------------------------------------------------------------------
Dim sql As String
Dim VarTexto As String
Dim Line As String
Dim Module As String
Dim i As Integer

Open "D:\pat\timetable\add_population_activities.txt" For Input As #1
Open "D:\pat\timetable\add_population_modules.txt" For Output As #2

sql = "select Title from module"
Adodc2.RecordSource = sql

Do Until EOF(1)
Line Input #1, VarTexto

Adodc2.Refresh
      Adodc2.Recordset.MoveFirst
      Do Until Adodc2.Recordset.EOF

            If Line = "REM Module: " + Adodc2.Recordset!title Then
               Module = Adodc2.Recordset!title

               Select Case Module
               Case "B31CI3" '3
                  Print #2, VarTexto
                  VarTexto = number_population_module(3)
                  Line Input #1, VarTexto

               Case "B31DI2" '109
                  Print #2, VarTexto
                  VarTexto = number_population_module(109)
                  Line Input #1, VarTexto

               Case "B31EI3" '128
                  Print #2, VarTexto
                  VarTexto = number_population_module (128)
                  Line Input #1, VarTexto

               Case "B31EN1" '105
                  Print #2, VarTexto
                  VarTexto = number_population_module (105)
                  Line Input #1, VarTexto
                        ……..
End Select
         End If
      Adodc2.Recordset.MoveNext
      Loop
      Adodc2.Recordset.Close
Print #2, VarTexto
Loop
Close
End Sub
```

Delete term 3

```
Private Sub Command2_Click()
'-------------------------------------------------------
'this program is to delete term 3 of module
'-------------------------------------------------------
Dim title As String
Dim i As Integer
Dim count As Integer
Dim j As Integer
Dim hostKey As String
Dim sql As String
Dim sql1 As String
Dim Line As String
Dim VarTexto As String
Dim myLine As String
Dim term3 As String
count = 1
Open "D:\pat\timetable\ add_population_modules.txt" For Input As #1
Open "D:\pat\timetable\delete_term3.txt" For Output As #2

 sql1 = "select Title, hostKey from module where Term_11063 =
'00000000000000000000000000001111111111000000000000000' OR Term_11063=
'00000000000000000000000000001111111111000000000000000'"
Adodc2.RecordSource = sql1
Adodc2.Refresh
Do Until EOF(1)
Line Input #1, VarTexto
i = nCaractersUntilEmpty(VarTexto, 1)
Line = Mid(VarTexto, 1, i) 'take id

Adodc2.Recordset.MoveFirst
Do Until Adodc2.Recordset.EOF

   If Line = "11000" Then
   myLine = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id
   j = nCaractersUntilEmpty(myLine, 2)
   hostKey = Mid(VarTexto, j, Len(VarTexto))
      If (Adodc2.Recordset!hostKey = hostKey) Then 'delete module
      Line Input #1, VarTexto
      End If
   End If

   title = Adodc2.Recordset!title
   If Line = "REM Module: " + title Then
     Line Input #1, VarTexto
     i = nCaractersUntilEmpty(VarTexto, 1)
     Line = Mid(VarTexto, 1, i)
     While Line <> "REM"
      Line Input #1, VarTexto
      i = nCaractersUntilEmpty(VarTexto, 1)
      Line = Mid(VarTexto, 1, i)
     Wend
   End If
Adodc2.Recordset.MoveNext
Loop
Print #2, VarTexto Loop
Adodc2.Recordset.Close
End Sub
```

Change name module and activity and change module pattern week

```
Private Sub Command3_Click()
'-------------------------------------------------------------------------------------------------------------------------------------------
'this program change module and activity name and module pattern week (use function isTerm1, isTerm2 and Opposite)
'-------------------------------------------------------------------------------------------------------------------------------------------
Dim VarTexto As String
Dim miLinia As String
Dim miLinia1 As String
Dim miLinia2 As String
Dim miLinia3 As String
Dim Line As String
Dim name As String
Dim count As Integer
Dim i As Integer
Dim j As Integer
Dim p As Integer
Dim k As Integer
Dim hostKey As String
Dim newLine As String
Dim title As String
Dim sql As String
Dim sql1 As String
Dim sign As String

Open "D:\pat\timetable\delete_term3.txt" For Input As #1 'file without module term 3
Open "D:\pat\timetable\change_name_terms_module.txt" For Output As #2

sql = "select old_code, new_code from module_codes"
Adodc2.RecordSource = sql
count = 1
Do Until EOF(1)

Line Input #1, VarTexto

i = nCaractersUntilEmpty(VarTexto, 1)
Line = Mid(VarTexto, 1, i) 'Take Id

If Line = "11004" Then
miLinia = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id
j = nCaractersUntilEmpty(miLinia, 2)
miLinia = Mid(miLinia, j + 2, Len(miLinia)) 'take name

Adodc2.Refresh
Adodc2.Recordset.MoveFirst
Do Until Adodc2.Recordset.EOF

      If miLinia = Adodc2.Recordset!old_code Then
        VarTexto = Mid(VarTexto, 1, nCaractersUntilEmpty(VarTexto, j + 3)) & Chr(9) & Adodc2.Recordset!new_code

      End If
  Adodc2.Recordset.MoveNext
Loop
Adodc2.Recordset.Close
End If
```

```
If Line = "12004" Then
miLinia1 = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id
j = nCaractersUntilEmpty(miLinia1, 2)
miLinia = Mid(miLinia1, j + 2, Len(miLinia1)) 'take name

Adodc2.Refresh
Adodc2.Recordset.MoveFirst
Do Until Adodc2.Recordset.EOF
   If (Mid(miLinia, 1, Len(Adodc2.Recordset!old_code)) = Adodc2.Recordset!old_code) Then 'change name of activity
         VarTexto = Mid(VarTexto, 1, nCaractersUntilEmpty(VarTexto, 1)) & Mid(miLinia1, 1, nCaractersUntilEmpty(miLinia1, j)) & Chr(9) & Adodc2.Recordset!new_code & Mid(miLinia, Len(Adodc2.Recordset!new_code), Len(miLinia))
   End If
   Adodc2.Recordset.MoveNext
Loop
Adodc2.Recordset.Close
End If
```

'term to semester

```
If Line = "11063" Then
miLinia = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id
j = nCaractersUntilEmpty(miLinia, 2)
miLinia = Mid(miLinia, j + 2, Len(miLinia)) 'take term


   If isTerm1(miLinia) Then
   newLine = "11111111111110000000000000000000000000000000000000"
   VarTexto = Mid(VarTexto, 1, nCaractersUntilEmpty(VarTexto, j + 3)) & Chr(9) & newLine
   ElseIf isTerm2(miLinia) Then
   newLine = "00000000000000000111111111111000111100000000000000"
   VarTexto = Mid(VarTexto, 1, nCaractersUntilEmpty(VarTexto, j + 3)) & Chr(9) & newLine
   End If

End If

If Line = "11081" Then

miLinia = Mid(VarTexto, i + 2, Len(VarTexto)) 'delete Id
p = nCaractersUntilEmpty(miLinia, 2)
hostKey = Mid(miLinia, 1, nCaractersUntilEmpty(miLinia, p - 1)) 'take hostkey
miLinia2 = Mid(miLinia, p + 2, Len(miLinia))
j = nCaractersUntilEmpty(miLinia2, 1)
sign = Mid(miLinia2, 1, j)
miLinia3 = Mid(miLinia2, j + 2, Len(VarTexto))
k = nCaractersUntilEmpty(miLinia3, 1)

If (sign <> "-all") Then
VarTexto = Mid(VarTexto, 1, nCaractersUntilEmpty(VarTexto, 1)) & Chr(9) & Mid(miLinia, 1, nCaractersUntilEmpty(miLinia, 1)) & Chr(9) & Mid(miLinia2, 1, nCaractersUntilEmpty(miLinia2, 1)) & Chr(9) & Opposite(newLine) & Mid(miLinia3, nCaractersUntilEmpty(miLinia3, k) + 1, Len(VarTexto))
End If

End If
Print #2, VarTexto
Loop
Close
End Sub
```

Change activity pattern week

```
Private Sub Command2_Click()
'-----------------------------------------------------------------------------------------------------------------------------------------
'this program change activity pattern week and line 11081 opposite to 12003 only in Lectures and Tutorials (use function isTerm1, isTerm2 and Opposite)
'-----------------------------------------------------------------------------------------------------------------------------------------
'Change term to semester Lectures and tutorial (hostKey)
Dim VarTexto As String
Dim miLinia As String
Dim miLinia1 As String
Dim miLinia2 As String
Dim miLinia3 As String
Dim i As Integer
Dim j As Integer
Dim p As Integer
Dim k As Integer
Dim s As Integer
Dim sql As String
Dim hostKey As String
Dim Line As String
Dim sign As String
Dim newLine As String

Open "D:\pat\timetable\change_name_terms_module.txt" For Input As #1 'file without module term3 and change name modules and activities
Open "D:\pat\timetable\terms_activity_final_file.txt" For Output As #2

Do Until EOF(1)
Line Input #1, VarTexto

i = nCaractersUntilEmpty(VarTexto, 1)
Line = Mid(VarTexto, 1, i) 'Take Id


If Line = "12063" Then
miLinia1 = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id
j = nCaractersUntilEmpty(miLinia1, 2)
miLinia = Mid(miLinia1, j + 2, Len(miLinia1)) 'take term

   If isTerm1(miLinia) Then
   newLine = "11111111111111000000000000000000000000000000000000000"
   VarTexto = Line & Mid(miLinia1, 1, nCaractersUntilEmpty(miLinia1, j)) & Chr(9) & newLine
   ElseIf isTerm2(miLinia) Then
   newLine = "00000000000000000111111111111000111100000000000000000"
   VarTexto = Line & Mid(miLinia1, 1, nCaractersUntilEmpty(miLinia1, j)) & Chr(9) & newLine
   Else
   newLine = miLinia
   VarTexto = Line & Mid(miLinia1, 1, nCaractersUntilEmpty(miLinia1, j)) & Chr(9) & newLine
   End If

End If
```

```
sql = "select HostKey from activity where ActivityTypes ='AL' OR ActivityTypes='T'"

If Line = "12081" Then
miLinia = Mid(VarTexto, i + 2, Len(VarTexto)) 'delete Id
p = nCaractersUntilEmpty(miLinia, 2)

hostKey = Mid(miLinia, 1, nCaractersUntilEmpty(miLinia, 1)) 'take hostkey
miLinia2 = Mid(miLinia, p + 2, Len(miLinia))

j = nCaractersUntilEmpty(miLinia2, 1)
sign = Mid(miLinia2, 1, j)

miLinia3 = Mid(miLinia2, j + 2, Len(VarTexto))
k = nCaractersUntilEmpty(miLinia3, 1)

Adodc2.RecordSource = sql
Adodc2.Refresh
Adodc2.Recordset.MoveFirst
Do Until Adodc2.Recordset.EOF

   If (hostKey = Adodc2.Recordset!hostKey) Then
     If (sign <> "-all") Then
       VarTexto = Mid(VarTexto, 1, nCaractersUntilEmpty(VarTexto, 1)) & Chr(9) & Mid(miLinia, 1, nCaractersUntilEmpty(miLinia, 1)) & Chr(9) & Mid(miLinia2, 1, nCaractersUntilEmpty(miLinia2, 1)) & Chr(9) & Opposite(newLine) & Mid(miLinia3, nCaractersUntilEmpty(miLinia3, k) + 1, Len(VarTexto))
   End If
   End If

Adodc2.Recordset.MoveNext
Loop
Adodc2.Recordset.Close

End If

Print #2, VarTexto
Loop
Close
End Sub
```

Functions

```
Function nCaractersUntilEmpty(sFileName As String, desde As Integer) As Integer
'------------------------------------------------------------
'this function strips the characters of the line
'------------------------------------------------------------
Dim i As Integer

For i = desde To Len(sFileName)
     If Mid(sFileName, i, 1) = Chr(9) Then          Chr(9) represents the space.
    Exit For
   End If
  Next
      nCaractersUntilEmpty = i - 1
End Function


Function number_population_activity(number As Integer)
'-------------------------------------------------------------------------------------------------------------------
'this function write number population of the parameters in line 12021 and 12039 to activity
'-------------------------------------------------------------------------------------------------------------------

Dim VarTexto As String
Dim Line As String
Dim i As Integer
Dim j As Integer
Dim Text As String
Dim myLine As String

            Line Input #1, VarTexto
            i = nCaractersUntilEmpty(VarTexto, 1)
            Line = Mid(VarTexto, 1, i) 'Take Id

            While Line <> "12021"
            Print #2, VarTexto
            Line Input #1, VarTexto
            i = nCaractersUntilEmpty(VarTexto, 1)
            Line = Mid(VarTexto, 1, i) 'Take Id
            myLine = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id
            j = nCaractersUntilEmpty(myLine, 2)
            Wend
            Text = Line & Mid(myLine, 1, j) & Chr(9) & number
            Print #2, Text

            Line Input #1, VarTexto
            While Line <> "12039"
            Print #2, VarTexto
            Line Input #1, VarTexto
            i = nCaractersUntilEmpty(VarTexto, 1)
            Line = Mid(VarTexto, 1, i) 'Take Id
            myLine = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id
            j = nCaractersUntilEmpty(myLine, 2)
            Wend
            Text = Line & Mid(myLine, 1, j) & Chr(9) & number
            Print #2, Text

End Function
```

```
Function number_population_module(number As Integer)
'-------------------------------------------------------------------------------------------------------------------
'this function write number population of the parameters in line 11021 and 11039 to activity.
-------------------------------------------------------------------------------------------------------------------

Dim VarTexto As String
Dim Line As String
Dim i As Integer
Dim j As Integer
Dim Text As String
Dim myLine As String

        Line Input #1, VarTexto
        i = nCaractersUntilEmpty(VarTexto, 1)
        Line = Mid(VarTexto, 1, i) 'Take Id

        While Line <> "12021"
        Print #2, VarTexto
        Line Input #1, VarTexto
        i = nCaractersUntilEmpty(VarTexto, 1)
        Line = Mid(VarTexto, 1, i) 'Take Id
        myLine = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id
        j = nCaractersUntilEmpty(myLine, 2)
        Wend
        Text = Line & Mid(myLine, 1, j) & Chr(9) & number
        Print #2, Text

        Line Input #1, VarTexto
        While Line <> "12039"
        Print #2, VarTexto
        Line Input #1, VarTexto
        i = nCaractersUntilEmpty(VarTexto, 1)
        Line = Mid(VarTexto, 1, i) 'Take Id
        myLine = Mid(VarTexto, i + 1, Len(VarTexto)) 'delete Id
        j = nCaractersUntilEmpty(myLine, 2)
        Wend
        Text = Line & Mid(myLine, 1, j) & Chr(9) & number
        Print #2, Text

End Function
```

```
Function isTerm1(Line As String) As Boolean
'-------------------------------------------------------------
'this function return true if the line is Term1
'-------------------------------------------------------------
Dim term1 As String
Dim i As Integer
i = 1
isTerm1 = True
term1 = Mid(Line, 11, Len(Line))

While i < Len(term1) And isTerm1
   If Mid(term1, i, 1) = "0" Then
      i = i + 1
      isTerm1 = True
   Else
     isTerm1 = False
   End If
Wend
End Function
```

```
Function isTerm2(Line As String) As Boolean
'-------------------------------------------------------------
'this function return true if the line is Term2
'-------------------------------------------------------------
Dim term1 As String
Dim term3 As String
Dim i As Integer
Dim j As Integer
i = 1
j = 1
isTerm2 = True
term1 = Mid(Line, 1, 14)
term3 = Mid(Line, 25, Len(Line))

While i < Len(term1) And isTerm2
   If Mid(term1, i, 1) = "0" Then
      i = i + 1

       isTerm2 = True
   Else
      isTerm2 = False
   End If
Wend
If isTerm2 Then
While j < Len(term3) And isTerm2
   If Mid(term3, j, 1) = "0" Then
      j = j + 1
      isTerm2 = True
   Else
      isTerm2 = False
   End If
Wend
End If
End Function
```

```
Function Opposite(sFileName As String) As String
'-------------------------------------------------------------
'this function changes the caracters 1 for 0 or opposite
'-------------------------------------------------------------
Dim i As Integer
Dim oFileName As String
oFileName = sFileName
i = 1
Do While i <= Len(sFileName)
   If Mid(sFileName, i, 1) = "0" Then
      Mid(oFileName, i, 1) = "1"
   ElseIf Mid(sFileName, i, 1) = "1" Then
      Mid(oFileName, i, 1) = "0"
   End If
   i = i + 1
Loop
     Opposite = oFileName
End Function
```