

CONVERTING FROM A LEGACY REPORTING SYSTEM TO DATA WAREHOUSE REPORTING UTILIZING SAS BI TOOLS

Kate Glover, Sinclair Community College, Dayton, OH

ABSTRACT

Successful projects don't just happen. They are the result of careful planning, negotiation, discovery and, perhaps most importantly, documentation. Requirements and specifications documents are the backbone of any project and cannot be overlooked.

INTRODUCTION

Sinclair Community College is a state funded educational institution. As such, there is a requirement to provide the state Board of Regents with comprehensive reporting on student and faculty demographics and enrollment, courses offered, alternative credit types, subsidy eligibility, etc. Because much of the college's funding is based on the information provided, accuracy and timeliness of reporting is critical.

The current processes include capture of all related data from the legacy system on both the 14th day of the term and the 30th day after the end of the term. After the 30th day capture, data is reconciled to select only those students who meet state qualifications for funding. After reconciliation is complete, reports are generated and submitted electronically to the state.

Currently, capture is via COBOL program. Captured files are printed and also transmitted electronically to the Research, Analytics, and Reporting department which, in turn, manually reconciles the data, verifies reports, and submits them to the Board of Regents in the form of Excel spreadsheets.

In early 2004, a decision was made to rewrite the existing capture and reporting processes to take advantage of SAS BI tools and the newly created data warehouse. The BI tools and the ability to report from the newly created data warehouse enables us to more easily enhance and modify reporting when there are changes mandated by the Board of Regents. It also improves our ability to provide data to Administration to support effective decision-making based on the freeze dates. The capture process will be written utilizing MS SQL while the reporting will be done through Enterprise Guide. This paper is intended to identify some of the important considerations in such a project.

One of the advantages of going to a Slowly Changing Dimensions Warehouse is that the reports for a given quarter can be run at any time after the appointed dates. In the legacy system, these reports had to be run at exactly the right time on the exact day. This caused a downtime to processing while the reports were run. If there is faulty data, the reports can not be recreated after correction; output reports had to be fixed manually. With the new system, we can create a datamart to reflect the data at a specific time in the past, and thus, there will be no need to bring down the Student Information System to run reports. This will provide a higher uptime rate and give us the ability to fix data in the system after the fact and then generate the reports.

CURRENT PROCESS

The purpose of the current process is to create one list of students who meet State requirements for funding by capturing and reconciling data frozen on two dates. Captured data is saved into two files. OB010FILE contains basic student information and information about up to 20 courses for which a student is registered on the freeze date. OB010MAST contains all the demographics about the students. Lists of variables for each file are included in Appendix A.

The two groups of students are reconciled according to a set of rules based on registration. These rules include:

- Pulling student demographics from the 14th day freeze captured on the 14th day after the beginning of the term, if a student is registered on the freeze date; from the 30th day freeze captured 30 days after the end of the term, if a student has a posted grade on the freeze date but was not on the 14th day capture; and from the 14th day freeze if a student is registered on the 14th day and has a grade posted on the 30th day freeze.

A student who is registered on the 14th day but has no grade posted on the 30th day freeze is dropped from the file.

- Because many courses are offered asynchronously – that is, they do not start on the traditional first day of the term – the Board of Regents has implemented a 20% census date rule. This means that a student is only considered “registered” for a course if they have registered by the date which represents 20% of the total length of the class. For traditional, full-term classes, that date is the 14th day of the term. However, for non-traditional short-term classes, the 20% census date has to be calculated from the first actual day of the class and not the first day of the term. To further complicate the issue, for classes which run less than 7 days – and there are some – the student has to be registered no later than the last day of the class.

One problem that Sinclair is wrestling with is that of special “bulk registrations”. These are students who are taking a restricted class and for which registration for the entire class is handled by a program coordinator. There have been problems with these registrations not being completed by the 14th day freeze. In that case, the College does not receive state funding for these students. Because these students were taking a full-term class but were not registered on the 20% census date – the 14th day freeze – even though they received grades by the 30th day freeze, they are dropped from files.

During summer terms, the process has another level of complexity as Sinclair recognizes four mini-terms, each of a different length of class. Each mini-term is processed as though it was a traditional term, but the 14th and 30th day reconciliation has to consider each mini-term as having a 14th day freeze. This is going to become a bigger issue as we move into mini-terms for more academic quarters.

NEW PROCESS

Sinclair currently has a transition process. On the capture dates, a back-up copy of the data warehouse is made and copied to the test environment. A special database was created for testing purposes. The OB010FILE and OB010MAST files generated by the current process are downloaded from the legacy system and copied into the test environment for comparison. Additionally, tables were created in the data warehouse to replicate the OB010FILE and OB010MAST files.

The transition process is running parallel with the existing process. A SQL script has been developed which is continually evolving as our understanding grows. The script is based on the list of variables in the OB010FILE and OB010MAST files and is almost organic in nature. You can see an example of the script in the “Validation of New Processes” section of this paper. It serves as both a testing tool and documentation of the process. Testing is being done each academic quarter; discrepancies are identified and investigated to determine cause of differences. After a full year’s testing without variance, the existing/legacy processes will be discontinued. We are not there yet.

REQUIREMENTS DOCUMENTATION

A basic rule of project management is to always have a requirements document before undertaking a project. There are lots of good examples on the Web; most of them differ from each other in one or more significant ways. However, it isn’t so much the form that you use; it’s the process of negotiation, discovery, documentation that matters. Without a requirements document, it’s impossible to know where the project is going and when it has arrived.

One of the common struggles many institutions have is that of starting a project implementation prior to the finish – or even the start - of a requirements document. Many times, as the Sinclair project team experienced, the requirements are “assumed” to be obvious and therefore do not require documentation. This is especially true if the project entails conversion from an existing system, and “just do the same thing” is all the requirements document there is.

As Sinclair found out the hard way, the devil is in the details or, in this case, the *problems* are in the *missing* details. Because there was a Cobol scripted document that was currently creating the frozen files, no discovery was conducted and no requirements document was developed; it was assumed that the project team could use that document for requirements. However, such was not the case. The process of understanding exactly how the current process functions has been tedious and time-consuming. In essence, the discovery has happened – IS happening – via the SQL script noted above.

A good requirements document is not written in a vacuum. The document should be a collaborative effort between the client and the team leader or other person authorized by management to develop the document. Until the requirements document has been written and approved, it will be difficult to determine what resources will need to be assigned to the project. The presence of a requirements document would have significantly reduced the time that this

project has taken – and it isn't finished yet.

It may occasionally be difficult to determine who the client – or owner of the process – is when planning a requirements document, particularly if the project did not originate with the owner. At Sinclair, the putative owner actually had not been involved in the development of the legacy system and did not fully understand all of the logic involved. In this case, the owner was actually more of a recipient of output than a creator of the process. This owner was only marginally involved in the new project as the decision to replace the legacy project was based on other business considerations not directly connected to the client.

If your business already has an approved format for discovery and requirements documentation AND you are using it consistently, great! If there is a format but you aren't using it, you really need to start doing so. Unfortunately, many corporate – and academic – cultures are not attuned to the processes involved in developing a requirements document. You may find yourself fighting an uphill battle but, if victorious, the payback in time and money saved will be worth the effort.

This paper is not intended to be a comprehensive lesson in project management. However, the following are some, but not all, of the items that should be included in the document.

BUSINESS OBJECTIVE

The first part of the requirements process is identification of the problem to be solved or need to be met or advantage to be realized; essentially, why we are undertaking this project. Will it generate competitive advantage? Will it support strategic goals? Does it replace a system that has outlived its usefulness? What is the business rationale for undertaking this project at this time? If we think of the requirements document like a newspaper article, this is the "why" of the story.

PROJECT OBJECTIVES

The document must take into consideration the entire project in both breadth and depth; it must be both specific and detailed. Without doing this homework, it will be impossible to determine what resources in time, people, and money will be required to successfully complete the project. It isn't sufficient to state that we "will replace current processes with new processes." What are the steps entailed in identifying what the current process does? How do we validate that those processes represent current business procedures? What will we have to do to create the new processes? Will users require training? Is that part of the project? If so, how much training? Whose responsibility will it be to develop user documentation or conduct training? Continuing the journalism analogy, the document will need to address the *what*, *who*, and *when* of all the steps in the project that will have attached resources.

WANTS VS. NEEDS

The document will detail what *must be delivered* from what *it would be nice to have*; in other words, this is the reality check. It's possible to add so many bells and whistles to a project that it will never be completed. The requirements document is the process by which both sides of the collaboration decide what features are necessary for the project to be considered a success. Defining these in the document will ensure there is no "scope creep" – or will at least help reduce creep.

In the case of Sinclair's project, since the goal was to replace an existing system with a new one, the current system became the starting point for collaboration. After reviewing the outputs of the current processes and meeting with the client – the business owner of the process – a list of items to be abandoned was developed (See Attachment B) A list of items to be added was also developed. During the project, as new items were identified that the client wanted to add, we were able to go back to this document to review the project scope with the client. In these cases, we agreed to maintain a list of "wants" that can be considered as enhancements to the new system after the project is completed.

The wants also have to be balanced against the costs. While it may be technically possible to deliver everything the client wants, the cost in both time and money may make it unfeasible. Generation of the requirements document is a process of negotiation.

WANTS VS. NEEDS REDUX

It's good to address methodology for handling the unexpected. During the project at Sinclair, it was found that the code in one part of the existing code appeared to contradict code in another part or that code was technically correct but was incomplete with regard to the business procedure being supported. In a few instances, the coded procedure

was simply wrong in that it returned inaccurate data.

Strictly speaking, the charge for the project was to duplicate the existing processes. Because of the critical importance of accuracy of reporting to the State, however, it did not make sense to simply duplicate the errors with the intent of fixing them as enhancements after project completion.

In these instances, the manager on the project team would consult with the owner of the information - the group with ultimate responsibility for reporting to the state – to determine whether to revise the existing process or to make necessary changes in the new process.

It's good to take steps to control creep, but sometimes business expediency takes precedence. It's important to be flexible and realistic.

SPECIFICATIONS

The specifications document may be part of the formal requirements document or it may be a separate document. In a very complex document, you may want to have separate specifications documents for different portions of the project. What is critically important is that you HAVE a specification document. You cannot successfully complete a project on a wing and a prayer. A great deal of time will be lost either in digging out the specifications piece meal or in revising the code to correct errors if you don't have detailed specifications in advance.

The document should include every bit of information that will be needed to code the new processes: source field and table names, any calculations that must be performed, the order in which processes must occur, etc. Document exactly what the deliverables are, what the processes are, what the final form must be. Discover and document everything that the developers will need to deliver the finished document as defined in the requirements documentation within the time frame allotted.

One method we used at Sinclair was mapping. At the beginning of the project, an Excel spreadsheet was created for both the OB010FILE and OB010MAST files listing all of the fields that would have to be captured for reporting. There were columns to list the location of each field in the legacy system and the corresponding location from the data warehouse. Also listed was any logic that would have to be applied in determining which value to pull from a field. Calculated fields were identified.

When mapping, be sure to map fields all the way back to their source; this is particularly important in the case of calculated fields. These fields may be called in the existing processes by an assigned name. It's important to document not just the name of the calculated field but exactly how the calculation is derived; i.e. what fields are inputs and what calculations are performed.

One of the issues we encountered was that many of the outputs of the legacy system are based on calculated fields. As part of the sequential updating of the legacy student information system, calculated fields were often rewritten in newer versions of the query language. This resulted, in some cases, in the calculated field being embedded in multiple layers of coding. Having detailed specifications that defined the necessary calculations in terms of their actual fields would have reduced the time spent tracking through these embedded layers.

As we neared the end of this phase of the project, a new spreadsheet was created with newly identified fields to be added to the output processes, new functions that have been created, and eliminating fields that it had been determined would be abandoned. The new document also did not carry forward information pertaining to the legacy system. It is a cleaner document and more user friendly. (See Appendix C)

The problem we experienced at Sinclair was that, since we were replacing an existing system, it was expected that the legacy code would be the specifications document. In essence, the project team was in the position of "reverse engineering" programming to determine the process in place and then going forward with new programming. The difficulty with that approach was that the legacy code was written in a computer language – COBOL – that many of our team did not understand at the time. In essence, we had a specifications document we couldn't use.

STAFFING

It is important that the team consists of member resources necessary for successful completion. In a project like this, such a team should include:

- A team leader charged with responsibility for overseeing the entire project, including the business process

- flow, technical requirements, budget, and status reporting;
- One or more programmers responsible for coding the new processes, and
- One or more persons who will conduct testing to determine that the new processes, when complete, could successfully replicate the results of the existing processes.

It's critical to identify, during the planning phase, what resources outside the basic project group may be necessary and to ensure that those resources are committed. For example, if all the members of the team are chosen from only one reporting group, will the team have all the skill sets needed for successful completion of the project?

Another key aspect of staffing is change. People move within the institution as well as leave the institution or there may be organizational realignment that may seriously impact the composition of the team. In the early stages of our project, one of the key team members moved from our department to another. This individual was an expert on the legacy system. Unfortunately, the vacant position was not filled, so it was necessary to start getting help from outside of our department. This assistance was on an "as available" basis; due to work load, it often took weeks to get responses to our questions.

About a year and a half into the project, the college went through reorganization, and we lost one of the primary code developers, a position which was also not filled. Because of these two losses and subsequent unfilled positions, it became necessary to rely on other resources outside the project to supplement the remaining project team.

VALIDATION OF NEW PROCESSES

Develop a test plan to validate that outputs of new processes are as expected. The test plan is built on the specifications, not on the final coding. The object is to test that the final process does what it is supposed to do, not that it is doing what it is coded to do.

At Sinclair, it was decided to run parallel reporting for a full academic year. A SQL script was developed to test outputs of new logic, formulas, formatting, etc. against the same data from the existing processes. Here is an example:

```
--Determine seat count, existing process
SELECT SSN
      ,CAST(CREDIT_HOUR_CT AS DECIMAL(15,1))/100
FROM dwwork.dbo.HEI_Course_Section_Work
WHERE Term = '2007FA'
AND OBOR = '14TH'
```

54,150 07-19-07

```
--Determine seat count, new process
USE HEI
SELECT STUDENT_ID_NB
      ,CREDIT_HOURS_EARNED_CT
FROM HEI.dbo.FROZEN_STUDENT_ACADEMIC_CREDIT
WHERE Term_Id = 391
      AND OBOR_Day_Id = 14
```

54,190 07-23-07

```
/*07-23-07 WE ARE OFF 40 NET RECORDS FOR SEVEN STUDENTS. SIX OF THE STUDENTS HAVE
STU_ACAD_CRED RECORDS FOR FRESH START. WE ARE INCLUDING THOSE RECORDS IN FROZEN
COUNT BUT OB010 DOES NOT COUNT THEM. THIS ACCOUNTS FOR SIX OF THE RECORDS. THE
REMAINING 34 RECORDS ARE PROFICIENCY GRADES POSTED FOR ONE STUDENT. OB010 IS ONLY SET
TO PUT OUT 20 RECORDS. IN ORDER TO KEEP ALL OF THE OB010 REPORTS FROM BOMBING, IT WAS
DECIDED TO DELETE THIS STUDENT FROM OB010 RECORDS. HE WILL HAVE TO BE MANUALLY ADDED BY
ANALYTICS WHEN THEY RECONCILE HEI REPORTS TO THE STATE. CONSIDER THIS RECONCILED.*/
```

You can see that, in the case of the project, reconciling the old and new systems is a matter of determining where the differences lie and then deciding which figure is accurate. In some instances, it can be resolved by further refinement of our understanding of current business processes. In other cases, we found that the existing processes were not in

alignment with business processes. Based on the seriousness of the misalignment and the severity of the impact on state reporting, occasionally the decision was made by the process owner to have the old process corrected. In other cases, the decision was made to not change existing processes, but to correct the errors in the new processes.

CONCLUSION

Many lessons have been learned during this project at Sinclair Community College. These include the importance of:

- Fleshing out a requirements document before beginning the project,
- Having a fully detailed specifications document or documents before beginning the project,
- Handling changing project requirements,
- Differentiating between client needs and wants,
- Maintaining sufficient qualified staffing resources,
- Negotiating to secure committed time from resources outside the project team, and
- Developing a test plan built on project specifications, not on coding, to validate that outputs of new processes are as expected.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Kate Glover
Sinclair Community College
444 West Third Street
Dayton OH 45402-1460
Work Phone: (937) 512-4599
Fax: (937) 512-2049
E-mail: kate.glover@sinclair.edu
Web: www.sinclair.edu

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

OB010MAST(

SSN
 NAME
 1st OLD NAME
 2nd OLD NAME
 ENTRANCE CODE
 ADDRESS
 CITY
 STATE
 ZIP
 PHONES
 EMERGENCY PHONES
 PRIVACY
 RESIDENCY STATUS
 COUNTY CODE
 STATE CODE
 RELEASE CODE
 TERM ENTERING
 BIRTH DATE
 SEX
 MARITAL STATUS
 U.S. CITIZEN
 PERMANENT RESIDENT
 FILE NUMBER
 ISSUE DATE
 VISA TYPE
 NATIVE COUNTRY
 RACE
 VETERAN
 VA BENEFITS
 FINANCIAL AID
 HIGH SCHOOL NAME
 HIGH SCHOOL CODE
 RECEIVED HS TRANSCR
 HIGH SCHOOL DIPLOMA
 DATE HS GRADUATION
 OTHER COLLEGES
 ID CARD DATE
 ENTERING MAJOR
 MAJOR CATEGORY
 PREV ACADEMIC STATUS
 CUR ACADEMIC STATUS
 CUM GPA
 TOT HOURS COMPLETED
 TOT HOURS ATTEMPTED
 TOT HOURS EARNED
 DISMISSALS

OB010FILE

SSN
 NAME
 ENROLLMENT_STATUS
 TERM_CREDIT_HOURS
 CUM_CREDIT_HOURS
 MAJOR
 STUDENT_RANK
 SEX
 RESIDENCY_STATUS
 STATE_CODE
 COUNTY_CODE
 BIRTH_YEAR

TOT ACCEPT ON PROB
 TOT READMIT ON PROB
 TOT CONT ON PROB
 FIRST TERM AT SCC
 LAST CUR NEXT TERM
 NEXT TERM REGISTERED
 TOT NO TERMS AT SCC
 NUM DEGREES
 NO OF DEGREE APPL
 TOT HOURS TRANSFER
 CURRENT MAJOR
 TOT HOURS REGISTERED
 DAY EVE WEEKEND
 REG CHANGE INDICATOR
 VALID ID CARD TERM
 HOURS CURRENT TERM
 STAFF DISC TERM
 DISCOUNT DEPT
 FALL ACADEMIC YR
 FALL HOURS REG
 WIN ACADEMIC YR
 WIN HOURS REG
 SPR ACADEMIC YR
 SPR HOURS REG
 SUM ACADEMIC YR
 SUM HOURS REG
 FA LETTER COUNT
 FA TOT HRS ATTEMPTED
 FA TOT HRS EARNED
 FA TERM HRS ATT
 FA TERM HRS EARNED
 FA 1st TERM LETTER
 FA 1st YEAR LETTER
 FA 2nd TERM LETTER
 FA 2nd YEAR LETTER
 FA 3rd TERM LETTER
 FA 3rd YEAR LETTER
 FA POSITION ON IND
 ADVISOR NUMBER
 FACULTY NUMBER
 SEL SERVICE NUMBER
 INT CODE 5
 TERM GPA
 OBOR COURSES
 END OF RECORD
 HEI_Term_Cd
 OBOR_Day_

MARITAL_STATUS
 INST_TRANS_FROM
 BRANCH
 RACE
 ACADEMIC_PERIOD
 SUBSIDY_CODE
 ON_OFF_CAMPUS
 NO_ON_CAMPUS
 NO_OFF_CAMPUS
 OBOR_COURSES
 HEI_TERM_CD
 OBOR_DAY_CD

**FIELDS CURRENTLY BEING CAPTURED IN 14/30 DAY FREEZE PROCESS WHICH WILL
NO LONGER BE CAPTURED**

	Comments from PPA**
BRANCH	Seems ok to exclude for now. Will this become relevant in near future???? We do use the off campus flag to identify the courses to be included in the OC file for HEI. I don't know if this is different, or if it was just determined that it need not be frozen. While we do need a field of this sort, I'm not sure that it would need to be part of the frozen file.
ON/OFF CAMPUS	
1st OLD NAME	OK - we don't use names anyway
2nd OLD NAME	OK - we don't use names anyway
PHONES	OK - we don't use phone #'s & we'd want most current not frozen anyway
EMERGENCY PHONES	OK - we don't use phone #'s & we'd want most current not frozen anyway
RELEASE CODE	Don't know what this is but probably ok
FILE NUMBER	Don't know what this is but probably ok
ISSUE DATE	OK - all 0's
VETERAN	OK - we don't use
VA BENEFITS	OK - we don't use
HIGH SCHOOL NAME	We do use this but maybe doesn't need to be in frozen file (might be nice to have tho)
HIGH SCHOOL CODE	We do use this but maybe doesn't need to be in frozen file (might be nice to have tho)
RECEIVED HS TRANSCR	I'm not sure. I thought we didn't collect HS transcripts anyway??? I don't think we use this for anything, shouldn't need to be frozen.
HIGH SCHOOL DIPLOMA	OK (Is always 0 or missing)
OTHER COLLEGES	We do use this but maybe doesn't need to be in frozen file
ID CARD DATE	Don't know what this is but probably ok
ENTERING MAJOR	OK, no need to freeze
MAJOR CATEGORY	OK -- N/A
PREV ACADEMIC STATUS	OK
TOT HOURS COMPLETED	OK (Is always 0 or missing)
DISMISSALS	OK
TOT ACCEPT ON PROB	OK
TOT READMIT ON PROB	OK
TOT CONT ON PROB	OK
LAST CUR NEXT TERM	OK -- N/A

** OK means it's all right to abandon.

APPENDIX B

HEI Mapping

Student_Id	Person_Id_Nb	DWView.dbo.Person_C	
Term_Id	Term_Id	DWView.dbo.Term_C	
OBOR_Day_Cd		DWView.dbo.Term_C	If run date = class start date + 14 days then '14th' else if term_end_Date + 30 = current date then '30th' else 'Error'
Student_Id_Nb	Person_Id_Nb	DWView.dbo.Person_C	
HEI_Term_Cd			
Social_Security_Nb	Social_Security_Nb	DWView.dbo.Person_C	
Student_Nm	Last_Nm, First_Nm Middle_Nm - 1st initial	DWView.dbo.Person_C	
Line_1_Ad	Line_1_Ad	DWView.dbo.Address_C	Uses preferred address id
Line_2_Ad	Line_2_Ad	DWView.dbo.Address_C	
City_Nm	City_Nm	DWView.dbo.Address_C	
State_Cd	State_Cd	DWView.dbo.Address_C	
Zip_Cd	Zip_Cd	DWView.dbo.Address_C	
Home_Phone_Nb	Phone_Nb	DWView.dbo.Address_Phone_C	Uses min(Counter_id) where Phone_Type_Cd = 'Home'
Work_Phone_Nb	Phone_Nb	DWView.dbo.Address_Phone_C	Uses min(Counter_id) where Phone_Type_Cd = 'Bus'
Emergency_Phone_Nb	Emergency_Contact_Phone_Nb	DWView.dbo.Person_C	Emergency_Contact_Phone_Nb IN PERSON_C
Privacy_In	privacy270_Person_privacy_in	DWView.dbo.Calc_Person_C	
Residency_Status_Cd	SR_Residency_Status_Cd	DWView.dbo.Student_Residency_C	Might have to validate this - DONE;
Residence_Count_y_Cd	CountyCode120_AddressCounty_Cd	DWView.dbo.CalcPerson_C	CORRECTED TO CALC
Residence_State_Cd	Residence_State_Cd	DWView.dbo.Person_C	CASE WHEN FP.RESIDENCE_STAT E_CD IS NULL THEN FA.State_Cd ELSE FP.RESIDENCE_STAT E_CD END
Start_Term_Cd	Calc_First_Str_Term_Id_Cd	DWView.dbo.Calc_Student_C	
Birth_Dt	Birth_Dt	DWView.dbo.Person_C	OB010Mast reports as xxxxxxxx; Person reports as xxxx-xx-xx 00:00:00.000. Requires substring to compare. See Asst Info tab.
Gender_Cd	Gender_Cd	DWView.dbo.Person_C	
US_Citizen_In	studentIsCitizen	DWView.dbo.Calc_Person_C	
ADDITIONAL ROWS DELETED FOR APPENDIX LENGTH			

APPENDIX C