



Executive Summary

By Garhet Wright

Description of the Medicare Part D Business

Medicare Part D is a prescription drug benefit that Certified Medicaid Services subsidizes for applicable health plans. Their tight regulations and requirements, make it essential for Part D Sponsors to be able to convert CMS Files and provide up to date reporting on a regular basis. As of today, CMS has not issued any tool to convert their files into manipulative data.

CMS Conversion/Reporting Tool

The System I have built has three major components

- 1) Conversion of CMS Files
- 2) Reporting with real time Enrollment Data
- 3) Reconciliation of CMS Files

Conversion of CMS Files

The system has built in the formats of various CMS files that are needed for CMS reporting. Once the file has been transferred into the shared file location, the system will allow the user to select the file from a file directory.

Reporting from Real Time Enrollment Data

The system will allow the user to access the Enrollment Data tables and create reporting based on the information available. There are two reports already built in, which are the “Enrollment Report” and the “LEP Report.” There is also a way to customize a report by using the SQL Assistance form.

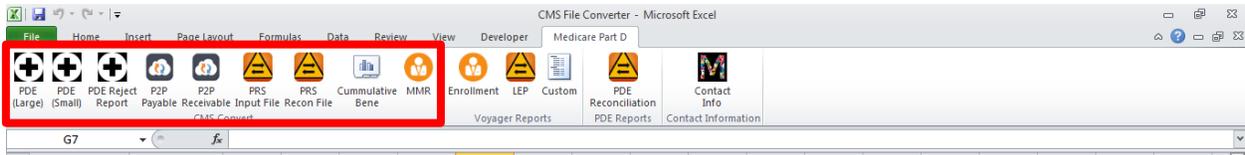
Reconciliation of CMS Files

Another benefit to the system is the ability to reconcile CMS Files. It will aggregate all of the necessary inputs and reconcile the accumulated amounts between reports.

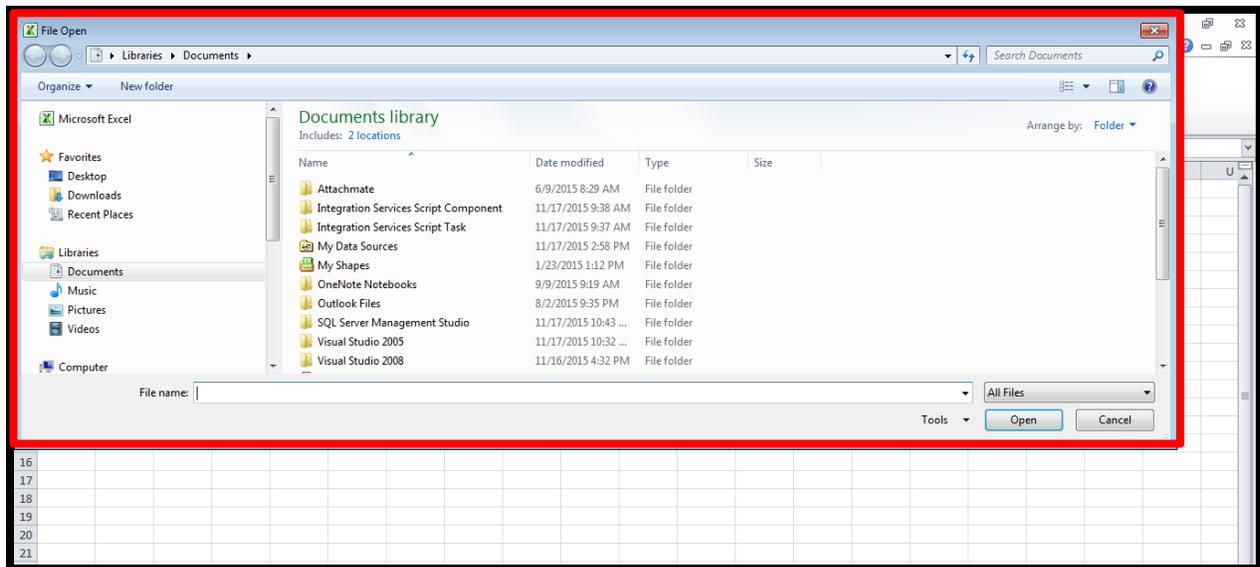
Implementation and Write Up

CMS Conversion Tool

The first thing I did was to enable the ability to choose which CMS file the user would like to convert. I customized the ribbon so that the user would be able to easily identify which file they will convert.



The first thing I did was to code the ability to choose from a file directory in order to locate the file that the user is hoping to convert. The icons represent at what level the data is being presented. For example, the first icon is representative of the information coming in at the claim level, the MMR and Enrollment information will come in at the beneficiary level, etc.



Once the user has elected a file to convert they will select the “Open” button and the code will begin to read the file line by line using a loop and Select Case function. The reason why a Select Case function is required is because there are some records that are unnecessary and do not need to be included. The Select Case determines which records should be read and which ones shouldn’t. The code will then format the file into an excel file with the appropriate headers.

	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	
1																						
2		PRESCRIBI	PRESCRIBI	DRUG COV	ADJUSTMI	NON- STA	PRICING E	CATASTR	INGREDIEI	SPENSIP	TOTAL AM	GROSS DR	GROSS DR	PATIENT F	OTHER TRI	LOW INCC	PATIENT L	COVERED	NON COV	ESTIMATE	VACCINE /	PRESC
3	1	Removed	C						\$0.72	\$1.30	\$0.00	\$2.02	\$0.00	\$0.33	\$0.00	\$0.00	\$0.00	\$0.00	\$1.69	\$0.00	\$0.00	
4	1	Removed	C						\$7.98	\$1.30	\$0.00	\$9.28	\$0.00	\$3.73	\$0.00	\$0.00	\$0.00	\$0.00	\$5.55	\$0.00	\$0.00	
5	1	Removed	C						\$18.57	\$2.25	\$0.00	\$20.82	\$0.00	\$2.55	\$0.00	\$5.45	\$0.00	\$0.00	\$12.82	\$0.00	\$0.00	
6	1	Removed	C						\$19.21	\$2.25	\$0.00	\$21.46	\$0.00	\$2.55	\$0.00	\$5.45	\$0.00	\$0.00	\$13.46	\$0.00	\$0.00	
7	1	Removed	C						\$21.02	\$2.25	\$0.00	\$23.27	\$0.00	\$2.55	\$0.00	\$5.45	\$0.00	\$0.00	\$15.27	\$0.00	\$0.00	
8	1	Removed	C						\$189.93	\$2.25	\$0.00	\$192.18	\$0.00	\$6.35	\$0.00	\$33.65	\$0.00	\$0.00	\$152.18	\$0.00	\$0.00	
9	1	Removed	C						\$6.98	\$2.25	\$0.00	\$9.23	\$0.00	\$2.00	\$0.00	\$0.00	\$0.00	\$0.00	\$7.23	\$0.00	\$0.00	
10	1	Removed	C						\$18.99	\$0.00	\$0.00	\$18.99	\$0.00	\$2.00	\$0.00	\$0.00	\$0.00	\$0.00	\$16.99	\$0.00	\$0.00	
11	1	Removed	C						\$4.73	\$2.25	\$0.00	\$6.98	\$0.00	\$2.00	\$0.00	\$0.00	\$0.00	\$0.00	\$4.98	\$0.00	\$0.00	
12	1	Removed	C						\$18.57	\$2.25	\$0.00	\$20.82	\$0.00	\$2.55	\$0.00	\$5.45	\$0.00	\$2.81	\$10.01	\$0.00	\$0.00	
13	1	Removed	C						\$152.25	\$1.00	\$0.00	\$153.25	\$0.00	\$6.35	\$0.00	\$33.65	\$0.00	\$0.00	\$113.25	\$0.00	\$0.00	
14	1	Removed	C						\$119.95	\$1.25	\$0.00	\$121.20	\$0.00	\$2.55	\$0.00	\$5.45	\$0.00	\$0.00	\$113.20	\$0.00	\$0.00	
15	1	Removed	C						\$16.71	\$1.00	\$0.00	\$17.71	\$0.00	\$2.00	\$0.00	\$0.00	\$0.00	\$0.00	\$15.71	\$0.00	\$0.00	
16	1	Removed	C						\$152.25	\$1.00	\$0.00	\$153.25	\$0.00	\$6.35	\$0.00	\$33.65	\$0.00	\$101.56	\$11.69	\$0.00	\$0.00	
17	1	Removed	C						\$1.17	\$1.25	\$0.00	\$2.42	\$0.00	\$2.00	\$0.00	\$0.00	\$0.00	\$1.81	(\$1.39)	\$0.00	\$0.00	
18	1	Removed	C						\$38.82	\$0.00	\$0.00	\$38.82	\$0.00	\$2.55	\$0.00	\$5.45	\$0.00	\$29.11	\$1.71	\$0.00	\$0.00	
19	1	Removed	C						\$119.95	\$1.25	\$0.00	\$121.20	\$0.00	\$2.55	\$0.00	\$5.45	\$0.00	\$90.90	\$22.30	\$0.00	\$0.00	
20	1	Removed	C						\$18.24	\$1.00	\$0.00	\$19.24	\$0.00	\$2.00	\$0.00	\$0.00	\$0.00	\$14.43	\$2.81	\$0.00	\$0.00	
21	1	Removed	C						\$152.25	\$1.00	\$0.00	\$153.25	\$0.00	\$6.35	\$0.00	\$33.65	\$0.00	\$114.94	(\$1.69)	\$0.00	\$0.00	
22	1	Removed	C						\$9.30	\$1.25	\$0.00	\$10.55	\$0.00	\$10.55	\$0.00	\$0.00	\$0.00	\$0.63	(\$0.63)	\$0.00	\$0.00	
23	1	Removed	C						\$32.89	\$1.25	\$0.00	\$34.14	\$0.00	\$8.00	\$0.00	\$0.00	\$0.00	\$0.00	\$26.14	\$0.00	\$0.00	
24	1	Removed	C						\$1.41	\$1.25	\$0.00	\$2.66	\$0.00	\$2.66	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
25	1	Removed	C						\$11.70	\$1.25	\$0.00	\$12.95	\$0.00	\$8.00	\$0.00	\$0.00	\$0.00	\$0.00	\$4.95	\$0.00	\$0.00	

Conversion of Currency Fields

A difficult portion of the code is the conversion of numerical fields. If CMS has reported a currency value, it is reported through an alphanumeric string of characters. For example, “0000000000123A” represents \$12.31. In order to convert the numeric fields, the code includes a loop that captures the entire string and separates the first 13 characters and evaluates the last character in order to determine the numerical value. It then puts it all together and formats the number as a currency.

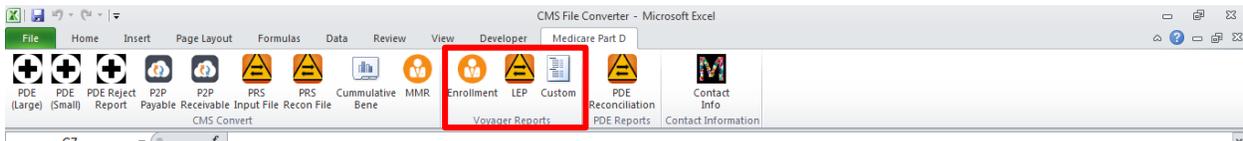
Vlookup with Unique Identifier for Contact Information

Another complication with the CMS files is that they provide a unique identifier but they do not provide all the needed information to process a payment. That information has to be cross-referenced with another file that contains contact information that CMS issues. In the system, the file will extract all of the needed information from the original CMS file and then cross-reference the unique identifier with another CMS file. I’ve illustrated the output of the conversion of a “P2P Payable File” below.

Invoice No	Member ID	P2P Amount	Contract Name	Last Name	First Name	Phone	E-Mail	Street Address	City	State	Zip
43COV20		\$145.41	METROPLUS HEALTH PLAN, INC.	Cocozza	Alisa	Alisa	cocozza@h	160 Water 4th Floor	New York	NY	10038
		\$34.15	TOUCHSTONE HEALTH HMO, INC.	Goryachk	Jenny	Jenny	ggoryachk	One North 12th Floor	White Pla	NY	10601
		\$116.52	TOUCHSTONE HEALTH HMO, INC.	Goryachk	Jenny	Jenny	ggoryachk	One North 12th Floor	White Pla	NY	10601
		\$259.56	THE NEW YORK STATE CATHOLIC HEALTH PLAN, INC.	Baker	Brian	Brian	bbaker@f	95-25 Que	0 Rego Park	NY	11374
		\$0.00	MANAGED HEALTH, INC.	Chen	Jin Zhi	Jin Zhi	jchen2@h	100 Churcl Attn: P2P	New York	NY	10007
		\$2.98	MANAGED HEALTH, INC.	Chen	Jin Zhi	Jin Zhi	jchen2@h	100 Churcl Attn: P2P	New York	NY	10007
		\$18.88	MANAGED HEALTH, INC.	Chen	Jin Zhi	Jin Zhi	jchen2@h	100 Churcl Attn: P2P	New York	NY	10007
		\$30.46	MANAGED HEALTH, INC.	Chen	Jin Zhi	Jin Zhi	jchen2@h	100 Churcl Attn: P2P	New York	NY	10007
		\$0.00	MANAGED HEALTH, INC.	Chen	Jin Zhi	Jin Zhi	jchen2@h	100 Churcl Attn: P2P	New York	NY	10007
		\$0.00	MANAGED HEALTH, INC.	Chen	Jin Zhi	Jin Zhi	jchen2@h	100 Churcl Attn: P2P	New York	NY	10007
		\$0.00	MANAGED HEALTH, INC.	Chen	Jin Zhi	Jin Zhi	jchen2@h	100 Churcl Attn: P2P	New York	NY	10007
		\$33.82	WELLCARE OF NEW YORK, INC.	Bolin	Elizabeth	Elizabeth	P2P-CASH	8735 Henc Attn: P2P	Tampa	FL	33634
		\$93.39	WELLCARE OF NEW YORK, INC.	Bolin	Elizabeth	Elizabeth	P2P-CASH	8735 Henc Attn: P2P	Tampa	FL	33634
		\$78.67	VNS CHOICE	Lee	Janet	Janet	Janet.Lee	1250 Broai 11th Floor	New York	NY	10001
		\$2.40	VNS CHOICE	Lee	Janet	Janet	Janet.Lee	1250 Broai 11th Floor	New York	NY	10001
		\$169.56	VNS CHOICE	Lee	Janet	Janet	Janet.Lee	1250 Broai 11th Floor	New York	NY	10001
		\$39.74	AMERIGROUP NEW YORK, LLC	Tran	Phuong	Phuong	Phuong.T	120 S. Via Mail Stop	Thousand	CA	91362
		\$0.00	AMERIGROUP NEW YORK, LLC	Tran	Phuong	Phuong	Phuong.T	120 S. Via Mail Stop	Thousand	CA	91362
		\$0.00	MEDCO CONTAINMENT LIFE INSURANCE COMPANY	Merzlicke	Carey	Carey	CMerzlick	One Expre Mailstop	Saint Loui	MO	63121
		\$4.93	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
		\$261.97	UNICARE LIFE & HEALTH INSURANCE COMPANY	Tran	Phuong	Phuong	Phuong.T	120 S. Via Mail Stop	Thousand	CA	91362
		\$52.01	WELLCARE PRESCRIPTION INSURANCE, INC.	Bolin	Elizabeth	Elizabeth	P2P-CASH	8735 Henc Attn: P2P	Tampa	FL	33634
		\$1,344.43									

Reporting with real time Enrollment Data

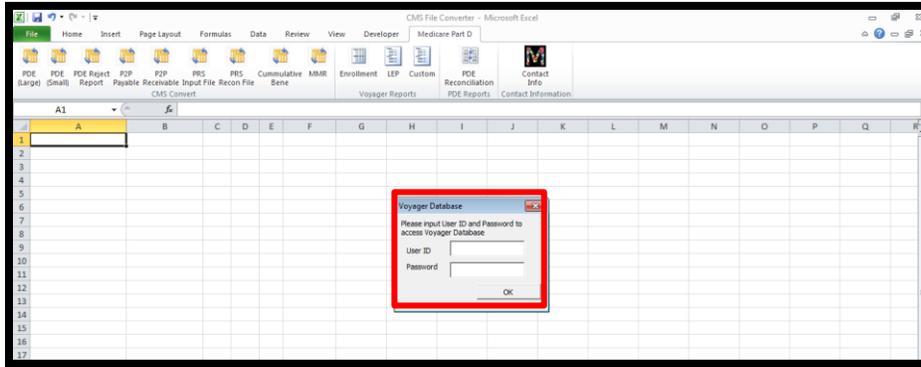
The ribbon has been customized in order to execute the three different options of Enrollment (Voyager) Reports.



The system has the ability to connect with the Enrollment Database and execute SQL queries in order to produce real time reporting. The system currently has three different options for reporting. The user can (1) select the Enrollment Report and receive an up to date report on how many members are currently enrolled in the plan and the source of their enrollment or (2) select the LEP Report that will pull a report that will show how many members have an outstanding Late Enrollment Penalty or (3) the user can build a custom report using the SQL Assistance form.

Connecting to the Database

The user will input the User ID, and Password that has been given to them by the Enrollment Vendor. This User ID and Password will be used to connect the system to the database.



Once the code has accessed the database it will run the query indicated below.

```
SELECT MEM.MEMBERID AS VOY_ID, MEM.MEDICAREID,
MEM.ENROLLMENTSOURCE, TRR.TRANSACTION_REPLY_CODE,
TRR.PLAN_CONTRACT_NUMBER, TRR.PLAN_BENEFIT_PACKAGE_ID,
TRR.EFFECTIVE_DATE, TRR.TRANSACTION_DATE,
TRR.COUNTY_CODE, TRR.STATE_CODE, MEM.ENROLLMENTDATE
FROM VYGR.MEMBER MEM
JOIN VYGR.CMS_TRANSACTION_REPLY TRR ON TRR.MEMBER_ID =
MEM.MEMBERID
WHERE TRR.TRANSACTION_REPLY_CODE IN (011, 022, 023, 100, 291)
AND MEM.STATUSCODE = 'ACTIVE'
```

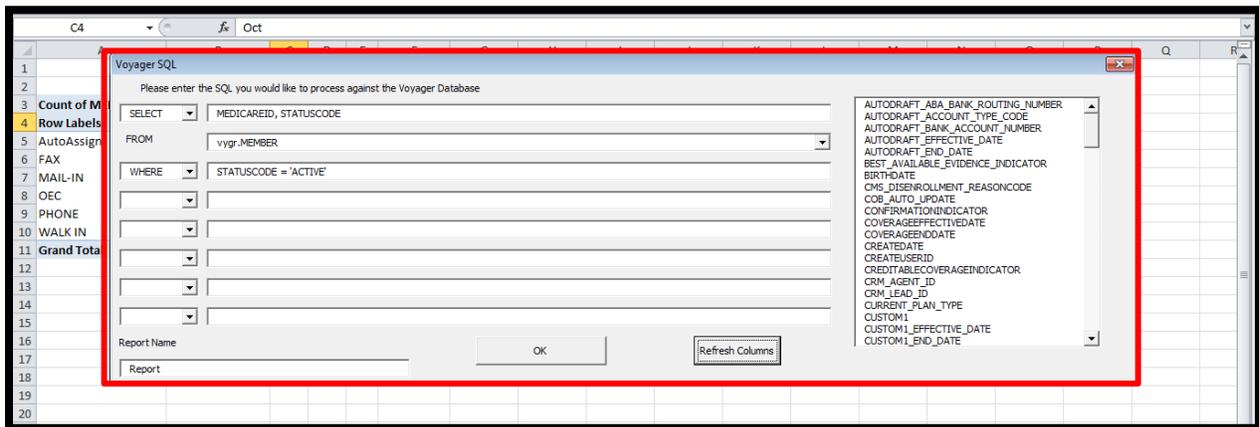
The code will then put all of the headers into the first row of the excel spreadsheet. It will then input all of the data that fits within the parameters into excel. This will be inputted into the created “Enrollment Data” tab. The code will then create a second tab to put a pivot table that will report the amount of enrollments each month and the source or those enrollments.

Count of MEDICAREID	Column Labels	Oct	Nov	Dec	Grand Total
AutoAssign	8581	2	784	936	
FAX	2	2			
MAIL-IN	14	33	4		
OEC	1	2101	9960	3391	1545
PHONE	273	1524	371	216	
WALK IN		1			
Grand Total	1	10971	11522	4546	2704

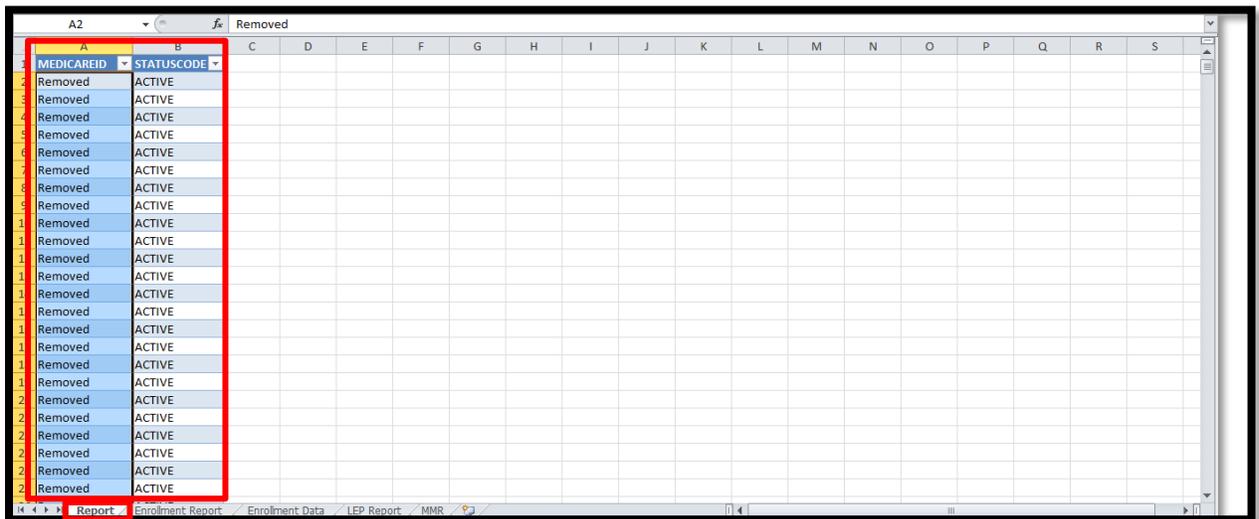
The only difference with the LEP Report is the parameters in the query. It also creates a separate tab labeled “LEP Report” and inputs all of the data into that spreadsheet.

Custom Voyager Report

If the user would like to use a custom report, they can use the SQL Assistance Form to create a report straight from the Enrollment Database.



With this form, the user can start to develop a query with the information they need. They would first choose a table that they would like to extract the data from. Once they have chosen the table, they can click on the “Refresh Columns” in order to identify all of the fields that are available in that table. From there, they can manually type in the fields they wish to pull from the data tables. The code takes all of the fields in order to develop the query to be executed. The user can also label the report by using the “Report Name” field. This name will show up on the tab the data is located as illustrated below.



Reconciliations of CMS Files

Once the CMS files have been appropriately converted, they have to be reconciled according to CMS regulations. The system will identify the appropriate fields in each file and place them into the correct place into the already formatted report. The report will then identify any variances between each file which will allow the user to report to CMS. After converting the file, the report will produce the output below.

PDE to Cumulative Beneficiary File Reconcile			
Data Field	PDE Data	Cumulative Beneficiary File	Difference
CLAIM_CT	965	965	0
INGRED_COST	\$ 69,208.57	\$ 69,208.57	\$0.00
DISP_FEE	\$ 1,465.95	\$ 1,465.95	\$0.00
TAX	\$ -	\$ -	\$0.00
GDCB	\$ 67,904.76	\$ 67,904.76	\$0.00
GDCA	\$ 2,769.76	\$ 2,769.76	\$0.00
TOTAL DRUG COST	\$70,674.52	\$ 70,674.52	\$0.00
PATIENT PAY	\$ 4,365.87	\$ 4,365.87	\$0.00
OTHER_TROOP	\$ 139.70	\$ 139.70	\$0.00
LICs_AMT	\$ 34,090.00	\$ 34,090.00	\$0.00
PLRO	\$ 260.87	\$ 260.87	\$0.00
CPP	\$ 25,636.55	\$ 25,636.55	\$0.00
NPP	\$ 6,181.53	\$ 6,181.53	\$0.00
VACCINE	\$ -	\$ -	\$0.00
GAP_DISCOUNT	\$ -	\$ -	\$0.00

Estimated PDE Reconciliation Amount			
Data Field	PDE Data	PDE Reconciliation File	Total

Contact Information

In the ribbon there is a button labeled “Contact Info.” When it is clicked, a form will be opened which will indicate who the head developer is and contact information if there is a question or request for an update.

Discussion of learning and conceptual difficulties encountered.

CMS Conversion

I had to learn how to add new tabs and delete tabs if they had the same name. It required using an Error Handling technique that I didn't previously know. After our error handling class, I was able to fix the problem rather quickly. There was also an issue with the combined files that I was using and I had to come up with an alternative solution for combined files that do not have the characters that indicate the end of the line of a text.

Reporting Real Time Enrollment Data

The only issue with this portion of the project was that when someone would enter in the right credentials but would then enter an invalid query, the code would not close the connection and reset it. So the next time I went to run it, the query would not run. So the solution I came up with is to create an if statement that would close the connection even if the query didn't work.

Reconciliation of CMS Files

There were no issues here. The primary lesson I learned was how to use the different Excel Application functions such as Vlookup and Sum within VBA. Other than that, there were no issues in developing the code.

Assistance

The only assistance I had was using code that has been published online and from the professor. I had an issue with converting text files earlier in the semester and I got his help on developing the code for it.

Conclusion

The appendix includes examples of the functionality of each area. Overall, the project was challenging but I was able to solutionize my own problems by either soliciting the help of the professor or the overly wise google. Thank you for a great semester!

Appendix 1: CMS File Conversion

```
Sub P2PPayableReport(button As Object)
report
```

```
End Sub
```

```
Sub report(Optional ByVal path As String)
Dim oneline As String
Dim P2PAmount As String
Dim s As Worksheet
Dim r As Long
```

```
Set s = Worksheets.Add
```

```
On Error Resume Next
Application.DisplayAlerts = False
Sheets("P2P Payable").Delete
Application.DisplayAlerts = True
On Error GoTo 0
```

```
s.Name = "P2P Payable"
```

```
If path = "" Then
    With Application.FileDialog(msoFileDialogOpen)
        .AllowMultiSelect = False
        .Show
        If .SelectedItems.Count = 0 Then Exit Sub
        path = .SelectedItems(1)
    End With
End If
```

```
Open path For Input As #1
Line Input #1, oneline
' we have the first line
```

```
DateofInvoice = CDate(Mid(oneline, 46, 2) & "/" & Mid(oneline, 48, 2) & "/" & Mid(oneline,
42, 4))
r = 1
```

```
s.Cells(1, 1).Value = "Invoice Number:"
s.Cells(1, 1).Font.Bold = True
s.Cells(1, 2).Value = Mid(oneline, 16, 16)
s.Cells(1, 3).Value = "Process Date"
s.Cells(1, 3).Font.Bold = True
s.Cells(1, 4).Value = DateofInvoice
s.Cells(2, 1).Value = "Member HICN"
```

```

s.Cells(2, 2).Value = "P2P Contract"
s.Cells(2, 3).Value = "P2P Amount"
s.Cells(2, 4).Value = "Contract Name"
s.Cells(2, 5).Value = "Last Name"
s.Cells(2, 6).Value = "First Name"
s.Cells(2, 7).Value = "Phone"
s.Cells(2, 8).Value = "E-Mail"
s.Cells(2, 9).Value = "Street Address1"
s.Cells(2, 10).Value = "Street Address2"
s.Cells(2, 11).Value = "City"
s.Cells(2, 12).Value = "State Code"
s.Cells(2, 13).Value = "Zip"
r = r + 1

```

Do Until EOF(1)

Line Input #1, oneline

Select Case left(oneline, 3)

Case "DET"

r = r + 1

s.Cells(r, 1).Value = Mid(oneline, 12, 20)

s.Cells(r, 2).Value = Mid(oneline, 122, 5)

s.Cells(r, 3).Value = convertCMSNumber(Mid(oneline, 127, 14))

s.Cells(r, 4).Value = "=VLOOKUP(" & s.Cells(r, 2).Address & ",Contacts!A:L,2,FALSE)"

s.Cells(r, 5).Value = "=VLOOKUP(" & s.Cells(r, 2).Address & ",Contacts!A:L,3,FALSE)"

s.Cells(r, 6).Value = "=VLOOKUP(" & s.Cells(r, 2).Address & ",Contacts!A:M,4,FALSE)"

s.Cells(r, 7).Value = "=VLOOKUP(" & s.Cells(r, 2).Address & ",Contacts!A:M,4,FALSE)"

s.Cells(r, 8).Value = "=VLOOKUP(" & s.Cells(r, 2).Address & ",Contacts!A:O,8,FALSE)"

s.Cells(r, 9).Value = "=VLOOKUP(" & s.Cells(r, 2).Address & ",Contacts!A:P,9,FALSE)"

s.Cells(r, 10).Value = "=VLOOKUP(" & s.Cells(r, 2).Address & ",Contacts!A:L,10,FALSE)"

s.Cells(r, 11).Value = "=VLOOKUP(" & s.Cells(r, 2).Address & ",Contacts!A:L,11,FALSE)"

s.Cells(r, 12).Value = "=VLOOKUP(" & s.Cells(r, 2).Address & ",Contacts!A:M,12,FALSE)"

s.Cells(r, 13).Value = "=VLOOKUP(" & s.Cells(r, 2).Address & ",Contacts!A:M,13,FALSE)"

Case "CTR"

r = r + 1

s.Cells(r, 3).Value = convertCMSNumber(Mid(oneline, 115, 14))

s.Cells(r, 2).Value = "Total P2P Amount"

s.Cells(r, 2).Font.Bold = True

End Select

```
Loop  
Close #1
```

```
Rows(2).Font.Bold = True
```

```
End Sub
```

```
Function convertCMSNumber(cmsNum As String) As Currency
```

```
Dim conv As String * 1
```

```
Dim pos As Integer
```

```
conv = Right(cmsNum, 1)  
If conv = "{" Then conv = "@"  
If Asc(conv) > 73 Then  
pos = -1  
If conv = "}" Then  
conv = "@"  
Else  
conv = Chr(Asc(conv) - 9)  
End If  
Else  
pos = 1  
End If
```

```
convertCMSNumber = (left(cmsNum, Len(cmsNum) - 1) & (Asc(conv) - 64)) * pos * 0.01
```

```
End Function
```

```
Sub P2PReconciliationPDE()
```

```
report2
```

```
End Sub
```

```
Sub report2(Optional ByVal path As String)
```

```
Dim oneline As String
```

```
Dim P2PAmount As String
```

```
Dim s As Worksheet
```

```
Dim r As Long
```

```
Set s = Worksheets.Add
```

```
On Error Resume Next
```

```
Application.DisplayAlerts = False
```

```
Sheets("P2P Reconciliation").Delete
```

```
Application.DisplayAlerts = True
On Error GoTo 0
```

```
s.Name = "P2P Reconciliation"
```

```
If path = "" Then
    With Application.FileDialog(msoFileDialogOpen)
        .AllowMultiSelect = False
        .Show
        If .SelectedItems.Count = 0 Then Exit Sub
        path = .SelectedItems(1)
    End With
End If
```

```
Open path For Input As #1
Line Input #1, oneline
```

```
DateofInvoice = CDate(Mid(oneline, 46, 2) & "/" & Mid(oneline, 48, 2) & "/" & Mid(oneline,
42, 4))
r = 1
```

```
s.Cells(1, 1).Value = "P2P Reconciliation for PDE Reconciliation"
s.Cells(2, 1).Value = "GDCB Amount"
s.Cells(2, 2).Value = "GDCA Amount"
s.Cells(2, 3).Value = "Total Gross Drug Cost"
s.Cells(2, 4).Value = "LICS Amount"
r = r + 1
```

```
Do Until EOF(1)
    Line Input #1, oneline
    Select Case left(oneline, 3)
        Case "CTR"
            r = r + 1
```

```
        s.Cells(r, 1).Value = convertCMSNumber(Mid(oneline, 37, 14))
        s.Cells(r, 2).Value = convertCMSNumber(Mid(oneline, 51, 14))
        s.Cells(r, 3).Value = convertCMSNumber(Mid(oneline, 65, 14))
        s.Cells(r, 4).Value = convertCMSNumber(Mid(oneline, 79, 14))
```

```
    End Select
Loop
Close #1
```

```
Rows(2).Font.Bold = True
```

Appendix 2 Real Time Enrollment Reporting

```

Option Explicit
Dim userid As String
Dim password As String
Dim cn As New ADODB.Connection
Dim rs As New ADODB.Recordset

Sub UserForm0
UserForm1.Show

End Sub

Sub UserForm20
frmSQL.Show

End Sub

Sub EnrollmentReport(button As Object)

    Dim r As Long
    Dim c As Long
    Dim s As Worksheet

    UserForm

    'userid = InputBox("User ID for Voyager Database", "Voyager User ID")
    'password = InputBox("Password for Voyager Database", "Voyager Password")

    userid = UserForm1.txbUserID.Value
    password = UserForm1.txbPassword.Value

    On Error GoTo ErrorHandler
    cn.Open "DRIVER=SQL Server;SERVER=209.161.38.207;DATABASE=VOYAGER",
userid, password
    rs.Open "SELECT MEM.MEMBERID AS VOY_ID, MEM.MEDICAREID,
MEM.ENROLLMENTSOURCE, TRR.TRANSACTION_REPLY_CODE,
TRR.PLAN_CONTRACT_NUMBER, TRR.PLAN_BENEFIT_PACKAGE_ID,
TRR.EFFECTIVE_DATE, TRR.TRANSACTION_DATE, TRR.COUNTY_CODE ,
TRR.STATE_CODE, MEM.ENROLLMENTDATE FROM VYGR.MEMBER MEM JOIN
VYGR.CMS_TRANSACTION_REPLY TRR ON TRR.MEMBER_ID = MEM.MEMBERID
WHERE TRR.TRANSACTION_REPLY_CODE IN (011, 022, 023, 100, 291) AND
MEM.STATUSCODE = 'ACTIVE'", cn

    Set s = Worksheets.Add

    On Error Resume Next
    Application.DisplayAlerts = False
    Sheets("Enrollment Data").Delete
    Application.DisplayAlerts = True
    On Error GoTo 0

```

```

s.Name = "Enrollment Data"

For c = 1 To rs.Fields.Count
    Sheets("Enrollment Data").Cells(1, c).Value = rs.Fields(c - 1).Name
Next c

r = 2
Do Until rs.EOF

    For c = 1 To rs.Fields.Count
        Sheets("Enrollment Data").Cells(r, c).Value = rs.Fields(c - 1).Value
    Next c
    r = r + 1
    rs.MoveNext
Loop

Range("a1").CurrentRegion.EntireColumn.AutoFit
ActiveSheet.ListObjects.Add(xlSrcRange, Range("a1").CurrentRegion, , xlYes).Name =
"Table 1"

rs.Close
cn.Close

'EnrollStates
EnrollPivottables

UserForm1.txbUserID.Value = ""
UserForm1.txbPassword.Value = ""

Exit Sub

ErrorHandler:
    MsgBox "Wrong UserID and/or Password. Please try again."
If Not (rs Is Nothing) Then
    If (rs.State And adStateOpen) = adStateOpen Then rs.Close
    Set rs = Nothing
End If
If Not (cn Is Nothing) Then
    If (cn.State And adStateOpen) = adStateOpen Then cn.Close
    Set cn = Nothing
End If

UserForm1.txbUserID.Value = ""
UserForm1.txbPassword.Value = ""

```

End Sub

Sub EnrollStates()

Dim x As Long

x = 2

Do Until IsEmpty(Sheets("Enrollment Data").Cells(x, 1))

Sheets("Enrollment Data").Cells(x, 12).Value =

Excel.Application.WorksheetFunction.VLookup(Sheets("Enrollment Data").Cells(x, 10).Value, Sheets("StateCodes").Range("A:B"), 2, False)

x = x + 1

Loop

Sheets("Enrollment Data").Cells(1, 12).Value = "State"

End Sub

Sub EnrollPivottables()

Dim s As Worksheet

Set s = Worksheets.Add

On Error Resume Next

Application.DisplayAlerts = False

Sheets("Enrollment Report").Delete

Application.DisplayAlerts = True

On Error GoTo 0

s.Name = "Enrollment Report"

ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:= _
Sheets("Enrollment Data").Range("A1").CurrentRegion,

Version:=xlPivotTableVersion14).CreatePivotTable TableDestination _

:=Sheets("Enrollment Report").Range("A3"), TableName:="PivotTable1",

DefaultVersion:= _

xlPivotTableVersion14

Sheets("Enrollment Report").Select

Cells(3, 1).Select

With ActiveSheet.PivotTables("PivotTable1").PivotFields("ENROLLMENTSOURCE")

.Orientation = xlRowField

.Position = 1

End With

With ActiveSheet.PivotTables("PivotTable1").PivotFields("ENROLLMENTDATE")

```

        .Orientation = xlColumnField
        .Position = 1
    End With
    ActiveSheet.PivotTables("PivotTable1").AddDataField ActiveSheet.PivotTables(_
        "PivotTable1").PivotFields("MEDICAREID"), "Count of MEDICAREID", xlCount
    Range("C4").Select
    Selection.Group Start:=True, End:=True, Periods:=Array(False, False, False, _
        False, True, False, False)
    'Range("A3").CurrentRegion.Select
    'Selection.Copy
    'Range("A3").End(xlDown).Offset(2, 0).Select
    'ActiveSheet.Paste
    'Range("A3").End(xlDown).Offset(2, 0).Name = "State"
    'ActiveSheet.PivotTables("State").PivotFields("ENROLLMENTSOURCE").Orientation =
xlHidden
    'With ActiveSheet.PivotTables("State").PivotFields("STATE_CODE")
        '.Orientation = xlRowField
        '.Position = 1
    'End With
    'Range("A3").CurrentRegion.Select
    'Selection.Copy
    'Range(Range("A3").End(xlDown).Offset(2, 0)).CurrentRegion.Offset(2, 0).Select
    'ActiveSheet.Paste
    'Range(Range("A3").End(xlDown).Offset(2, 0)).CurrentRegion.Offset(2,
0).PivotTable.Name = "PBP"
    'ActiveSheet.PivotTables("PBP").PivotFields("ENROLLMENTSOURCE"). _
        ' Orientation = xlHidden
    'With ActiveSheet.PivotTables("PBP").PivotFields(_
        ' "PLAN_BENEFIT_PACKAGE_ID")
        ' .Orientation = xlRowField
        ' .Position = 1
    'End With
End Sub

```

```

Sub LEPReport(button As Object)

```

```

    Dim r As Long
    Dim c As Long
    Dim s As Worksheet

```

```

    UserForm

```

```

    'userid = InputBox("User ID for Voyager Database", "Voyager User ID")
    'password = InputBox("Password for Voyager Database", "Voyager Password")

```

```

    userid = UserForm1.txbUserID.Value
    password = UserForm1.txbPassword.Value

```

```

On Error GoTo ErrorHandler
cn.Open "DRIVER=SQL Server;SERVER=209.161.38.207;DATABASE=VOYAGER",
userid, password
rs.Open "SELECT MEDICAREID, LEP_AMOUNT, LEP_EFFECTIVE_DATE FROM
VYGR.MEMBER WHERE STATUSCODE='ACTIVE' AND LEP_AMOUNT>0", cn
Set s = Worksheets.Add

```

```

On Error Resume Next
Application.DisplayAlerts = False
Sheets("LEP Report").Delete
Application.DisplayAlerts = True
On Error GoTo 0

```

```
s.Name = "LEP Report"
```

```

For c = 1 To rs.Fields.Count
    Sheets("LEP Report").Cells(1, c).Value = rs.Fields(c - 1).Name
Next c

```

```

r = 2
Do Until rs.EOF

```

```

    For c = 1 To rs.Fields.Count
        Sheets("LEP Report").Cells(r, c).Value = rs.Fields(c - 1).Value
    Next c
    r = r + 1
    rs.MoveNext
Loop

```

```

Range("a1").CurrentRegion.EntireColumn.AutoFit
ActiveSheet.ListObjects.Add(xlSrcRange, Range("a1").CurrentRegion, , xlYes).Name =
"Table 1"

```

```

rs.Close
cn.Close

```

```

UserForm1.txbUserID.Value = ""
UserForm1.txbPassword.Value = ""

```

```
Exit Sub
```

```
ErrorHandler:
```

```

MsgBox "Wrong UserID and/or Password. Please try again."
If Not (rs Is Nothing) Then
    If (rs.State And adStateOpen) = adStateOpen Then rs.Close

```

```
Set rs = Nothing
End If
If Not (cn Is Nothing) Then
    If (cn.State And adStateOpen) = adStateOpen Then cn.Close
    Set cn = Nothing
End If
```

```
UserForm1.txbUserID.Value = ""
UserForm1.txbPassword.Value = ""
```

```
End Sub
```

```
Sub VoyagerReport(button As Object)
```

```
    Dim r As Long
    Dim c As Long
    Dim s As Worksheet
    Dim sql As String
    Dim reportname As String
```

```
    UserForm
```

```
'userid = InputBox("User ID for Voyager Database", "Voyager User ID")
'password = InputBox("Password for Voyager Database", "Voyager Password")
```

```
userid = UserForm1.txbUserID.Value
password = UserForm1.txbPassword.Value
```

```
    UserForm2
```

```
With frmSQL
```

```
    sql = .cmbSQLSelect.Value & " " & .txtSQL1.Value & " " & "FROM" & " "
    & .cmbFrom.Value & " " & .cmbSQL1.Value & " " & .txtSQL3.Value
    reportname = .txtreportname.Value
```

```
End With
```

```
On Error GoTo ErrorHandler
    cn.Open "DRIVER=SQL Server;SERVER=209.161.38.207;DATABASE=VOYAGER",
    userid, password
    rs.Open sql, cn
    Set s = Worksheets.Add
```

```
On Error Resume Next
Application.DisplayAlerts = False
Sheets(reportname).Delete
Application.DisplayAlerts = True
On Error GoTo 0
```

```
s.Name = reportname
```

```
For c = 1 To rs.Fields.Count
    Sheets(reportname).Cells(1, c).Value = rs.Fields(c - 1).Name
Next c
```

```
r = 2
Do Until rs.EOF
```

```
    For c = 1 To rs.Fields.Count
        Sheets(reportname).Cells(r, c).Value = rs.Fields(c - 1).Value
    Next c
    r = r + 1
    rs.MoveNext
Loop
```

```
Range("a1").CurrentRegion.EntireColumn.AutoFit
ActiveSheet.ListObjects.Add(xlSrcRange, Range("a1").CurrentRegion, , xlYes).Name =
"Table 1"
```

```
rs.Close
cn.Close
```

```
UserForm1.txbUserID.Value = ""
UserForm1.txbPassword.Value = ""
```

```
Exit Sub
```

```
ErrorHandler:
```

```
    MsgBox "Wrong UserID and/or Password. Please try again."
If Not (rs Is Nothing) Then
    If (rs.State And adStateOpen) = adStateOpen Then rs.Close
    Set rs = Nothing
End If
If Not (cn Is Nothing) Then
    If (cn.State And adStateOpen) = adStateOpen Then cn.Close
    Set cn = Nothing
End If
```

```
UserForm1.txbUserID.Value = ""  
UserForm1.txbPassword.Value = ""
```

```
End Sub
```

Appendix 3: PDE Reconciliatio

```
Sub PDEReconciliation(button As Object)
```

```
Dim x As Byte
```

```
Dim s As Worksheet
```

```
Dim s2 As Worksheet
```

```
Set s = Worksheets.Add
```

```
On Error Resume Next
```

```
Application.DisplayAlerts = False
```

```
Sheets("PDE Reconciliation").Delete
```

```
Application.DisplayAlerts = True
```

```
On Error GoTo 0
```

```
s.Name = "PDE Reconciliation"
```

```
Sheets("PDEReconTemplate").Range("B2:E39").Copy
```

```
s.Paste
```

```
s.Range("a1").EntireColumn.Insert
```

```
s.Columns("B:E").ColumnWidth = 20
```

```
s.Range("C3").Value =
```

```
Excel.Application.WorksheetFunction.Count(Sheets("PDE").Range(Sheets("PDE").Range("A3"), Sheets("PDE").Range("A3").End(xlDown)))
```

```
s.Range("C4").Value =
```

```
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("A B3"), Sheets("PDE").Range("AB3").End(xlDown)))
```

```
s.Range("C5").Value =
```

```
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("A C3"), Sheets("PDE").Range("Ac3").End(xlDown)))
```

```
s.Range("C6").Value =
```

```
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("Ad 3"), Sheets("PDE").Range("Ad3").End(xlDown)))
```

```
s.Range("C7").Value =
```

```
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("Ae 3"), Sheets("PDE").Range("Ae3").End(xlDown)))
```

```
s.Range("C8").Value =
```

```
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("Af 3"), Sheets("PDE").Range("Af3").End(xlDown)))
```

```
s.Range("C10").Value =
```

```
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("A G3"), Sheets("PDE").Range("Ag3").End(xlDown)))
```

```
s.Range("C11").Value =
```

```
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("Ah 3"), Sheets("PDE").Range("Ah3").End(xlDown)))
```

```
s.Range("C12").Value =
```

```
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("AI 3"), Sheets("PDE").Range("Ai3").End(xlDown)))
```

```

s.Range("C13").Value =
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("AJ
3"), Sheets("PDE").Range("Aj3").End(xlDown)))
s.Range("C14").Value =
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("A
K3"), Sheets("PDE").Range("Ak3").End(xlDown)))
s.Range("C15").Value =
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("AL
3"), Sheets("PDE").Range("Al3").End(xlDown)))
s.Range("C16").Value =
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("A
N3"), Sheets("PDE").Range("An3").End(xlDown)))
s.Range("C17").Value =
Excel.Application.WorksheetFunction.Sum(Sheets("PDE").Range(Sheets("PDE").Range("A
W3"), Sheets("PDE").Range("Aw3").End(xlDown)))
s.Range("C9").Value = s.Range("c8").Value + s.Range("C7").Value

```

```

s.Range("D3").Value = Sheets("Cum Bene").Range("f2").End(xlDown)
s.Range("D4").Value = Sheets("Cum Bene").Range("g2").End(xlDown)
s.Range("D5").Value = Sheets("Cum Bene").Range("h2").End(xlDown)
s.Range("D6").Value = Sheets("Cum Bene").Range("i2").End(xlDown)
s.Range("D7").Value = Sheets("Cum Bene").Range("j2").End(xlDown)
s.Range("D8").Value = Sheets("Cum Bene").Range("k2").End(xlDown)
s.Range("D9").Value = Sheets("Cum Bene").Range("l2").End(xlDown)
s.Range("D10").Value = Sheets("Cum Bene").Range("m2").End(xlDown)
s.Range("D11").Value = Sheets("Cum Bene").Range("n2").End(xlDown)
s.Range("D12").Value = Sheets("Cum Bene").Range("o2").End(xlDown)
s.Range("D13").Value = Sheets("Cum Bene").Range("q2").End(xlDown)
s.Range("D14").Value = Sheets("Cum Bene").Range("r2").End(xlDown)
s.Range("D15").Value = Sheets("Cum Bene").Range("s2").End(xlDown)
s.Range("D16").Value = Sheets("Cum Bene").Range("ac2").End(xlDown)
s.Range("D17").Value = Sheets("Cum Bene").Range("ad2").End(xlDown)

```

For x = 1 To 15

```
s.Cells(x + 2, 5).Value = s.Cells(x + 2, 4).Value - s.Cells(x + 2, 3).Value
```

Next x

```

s.Range("c20").Value = s.Range("c7").Value
s.Range("c21").Value = s.Range("c8").Value
s.Range("c22").Value = s.Range("c9").Value
s.Range("c23").Value = s.Range("c12").Value
s.Range("D20").Value = Sheets("P2P Reconciliation").Range("A3").Value
s.Range("D21").Value = Sheets("P2P Reconciliation").Range("b3").Value
s.Range("D22").Value = Sheets("P2P Reconciliation").Range("c3").Value
s.Range("D23").Value = Sheets("P2P Reconciliation").Range("d3").Value

```

For x = 1 To 4

```
s.Cells(x + 19, 5) = Excel.Application.WorksheetFunction.Sum(s.Cells(x + 19, 3).Value,  
s.Cells(x + 19, 4).Value)
```

Next x

```
s.Range("E24").Value = InputBox("Please put an approximate DIR Amount to calculate the  
estimated CMS Reconciliation payment")
```

```
s.Range("E25").Value = s.Range("E21").Value / s.Range("E22").Value  
s.Range("E26").Value = (s.Range("E21").Value - (s.Range("E24").Value *  
s.Range("E25").Value)) * 0.8  
s.Range("e27").Value = s.Range("e23").Value  
s.Range("e28").Value = s.Range("e27").Value + s.Range("e26").Value
```

```
s.Range("E32").Value = Excel.Application.WorksheetFunction.Count(Sheets("PDE  
REJ").Range(Sheets("PDE REJ").Range("A3"), Sheets("PDE  
REJ").Range("A3").End(xlDown)))  
s.Range("E34").Value = Excel.Application.WorksheetFunction.Sum(Sheets("PDE  
REJ").Range(Sheets("PDE REJ").Range("Af3"), Sheets("PDE  
REJ").Range("Af3").End(xlDown)))  
s.Range("E35").Value = Excel.Application.WorksheetFunction.Sum(Sheets("PDE  
REJ").Range(Sheets("PDE REJ").Range("AI3"), Sheets("PDE  
REJ").Range("Ai3").End(xlDown)))  
s.Range("E36").Value = Excel.Application.WorksheetFunction.Sum(Sheets("PDE  
REJ").Range(Sheets("PDE REJ").Range("AW3"), Sheets("PDE  
REJ").Range("Aw3").End(xlDown)))
```

```
s.Range("E31").Value = s.Range("c3").Value + s.Range("e32").Value  
s.Range("E33").Value = 1 - (s.Range("E32").Value / s.Range("E31").Value)  
s.Range("E37").Value = Excel.Application.WorksheetFunction.Sum(s.Range("E34"),  
s.Range("E35"), s.Range("e36"))
```

End Sub