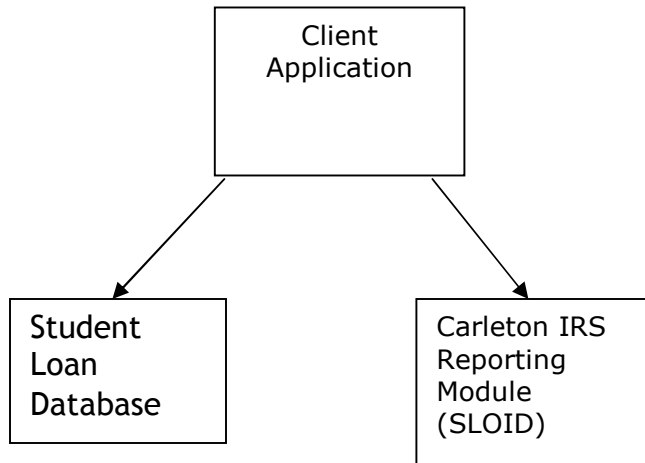


Carleton IRS Student Loan Reporting Module Implementation

Architecture

GENERAL ARCHITECTURE

- Client application pulls transaction data from processing database.
- Client application formats and passes information to Carleton IRS Reporting module for each payment stream. Carleton calculations AIP and OID and passes this information back to calling application for storage.



The Carleton module was created to perform the calculations required for reporting purposes for the 1098-E. The module was created based on the report 1098-E Reporting Under the Final 221 Regulations provided by Mary Doyle, Vice President Tax and Accounting, for the Sallie Mae Corporation.

In this report, detail is provided on the definitions of OID, AIP, and the Constant Interest Method calculations that need to be performed to compute the OID accrual for each period.

The Carleton IRS Reporting module is written in C. In an AS400 environment, the module accepts parameters that are character strings. The first string contains all of the input information from the InputRec definition listed below. The second string contains the output information as defined by OutputRec. This is represented by the sample RPG call below where PARMDS represents the input string and PARMOS represents the output string.

```
C          CALL 'SLOID'  
C          PARM          PARMDS  
C          PARM          PARMOS
```

The Carleton SLOID module parses the input string, performs the calculations, then concatenates all of the output data into the output string to pass back to the calling application.

The Carleton module would generate the following each period:

- 1) Yield to Maturity (YTM)
- 2) Yield Charge

Carleton IRS Student Loan Reporting Module Implementation

- 3) AIP
- 4) OID (Unpaid and Paid)

Computing AIP/OID

The module is called for each period within a student loan to compute the AIP and OID (paid and unpaid). The AIP computed from the prior period and any unpaid OID is passed back in on succeeding calls.

The fields needed to compute AIP/OID are:

- AIP
- QSI
- ThisPay
- Unpaid OID
- Pay[]
- PayNP[]
- LastPayDate Month,Day,Year
- ThisPayDate Month,Day,Year
- DueDate Month,Day,Year

All other fields can be set to zero. ***On an AS400 all unused fields should be padded with zeros.***

The yield charge for the period is computed as follows:

$$\text{Yield Charge} = \text{QSI} + \text{OID Unpaid} + \text{OID Paid}$$

The Yield to Maturity is computed internally and is not passed back as a parameter. The module always recalculates the YTM each time it is called. This eliminates the need to identify “re-yielding events.”

Note that capitalized (“capped”) interest is never input *per se*; it is built into the payments. We will need to know the conversion date (capping date) if the module must compute the payment streams.

Computing AIP/OID and re-computing Payment Streams

The module can calculate recalculate the payments streams by providing the following variables:

- AIP
- QSI
- ThisPay
- Principal
- Unpaid Interest
- Unpaid OID
- Pay[]
- PayNP[]
- Rate[]
- RateNP[]
- LastPayDate Month,Day,Year
- ThisPayDate Month,Day,Year
- DueDate Month,Day,Year
- ConversionDate Month,Day,Year
- InterestHandling

If the variable Principal is greater than 0 and Pay[0] is zero, the module calculates all streams.

M&D/365 is used for stream calculation along with the Interest Handling method. Available Interest Handling methods are as follows:

Carleton IRS Student Loan Reporting Module Implementation

- 0 = U.S. Rule (Escrow interest)
- 1 = Monthly Actuarial - Compound interest every month
- 2 = Escrow interest until Conversion Date, then compound
- 3 = Government pays interest until conversion, then U.S. Rule
- 4 = Government pays interest until conversion, then compound
- 5 = Escrow interest until Conversion Date, then U.S. Rule

Methods 3 and 4 should probably not be used; instead, use an interest rate of 0 before conversion.

Payment streams should start with the first *scheduled* payment due (DueDate) *after* LastPayDate, and ThisPay is ignored for the purposes of yield calculation. N.B.: **Even if the true due date is *on or before* LastPayDate, DueDate must be *strictly after* LastPayDate, since the yield-to-maturity calculation is purely forward-looking.**

Computing AIP/OID and recomputing Term and Final Payment

The module can calculate recalculate the payments streams by providing the following variables:

- AIP
- QSI
- ThisPay
- Principal
- Unpaid Interest
- Unpaid OID
- Pay[]
- PayNP[]
- Rate[]
- RateNP[]
- LastPayDate Month,Day,Year
- ThisPayDate Month,Day,Year
- DueDate Month,Day,Year
- ConversionDate Month,Day,Year
- InterestHandling

If the variable Principal is greater than 0 and Pay[0] is zero, the module calculates all streams.

If Pay[0] is greater than 0, the module will recalculate the term & final payment, using the payment streams provided.

Payment streams should start with the first *scheduled* payment due (DueDate) *after* LastPayDate. ThisPay is ignored for the purposes of yield calculation, but is only used to determine how much of the earned yield is paid and how much is unpaid.

OID Definition

MODULE CALL

Windows

```
void OID(InputRec, OutputRec)
```

AS400 RPG Call

Carleton IRS Student Loan Reporting Module Implementation

```

C          CALL 'SLOID'
C          PARM          PARMDS
C          PARM          PARMOS
    
```

*A sample RPG program will be provided.

AS400 COBOL Call

```

CALL "SLOID"
          USING PARMDS
          PARMOS
    
```

INPUTREC DEFINITION

* For field lengths below, decimal places are implied on AS400 or Mainframe systems. Field lengths are defined as x.y where x represents the number of characters preceding the decimal point and y represents the number of characters following the decimal point. 7.2 would be formatted as follows 9999999.99. When inputting negative numbers (for ThisPay or RateNP), put the minus sign in the first position, followed by the zero padding (e.g., "-01" for RateNP[0] to indicate a rate used until conversion).

NOTE: On AS400 systems, the input values will be concatenated together in a string format.

Field Length	Field Type	Field Name	Description
7.2	Double	AIP	Adjusted Issue Price. On the first call to the module, this is the cash disbursed to the student minus the origination fee. On succeeding calls, it is the AIP that was passed back from the last call to the module.
7.2	Double	QSI	Qualified Stated Interest. Amount of payment that went towards interest. Leave this zero to calculate the total yield charge instead of OID only.
7.2	Double	ThisPay	Amount of the actual payment that was applied to principal and interest only. This can not include late charges, NSF charges, etc. For disbursements <i>after</i> origination, input the amount of new AIP as a negative number.
7.2	Double	Principal	Principal amount after previous payment. <i>This field is used for payment stream recalculations only.</i> Recalculate streams when in the pre-payment period and do not have a payment yet, or before all disbursements are complete, or because the payments need to be recast because a large payment has occurred or many payments were missed.
7.2	Double	UnpaidInterest	Earned, unpaid interest after previous payment. <i>This field is used for payment stream recalculations only.</i> Use this in combination with the Principal when there is also outstanding unpaid interest (before interest capitalization, for example).
7.2	Double	UnpaidOID	Value from previous call from module for last period. The first call to the module on loans started after 1/1/2005 will contain 0. Any loans that have a payment

Carleton IRS Student Loan Reporting Module Implementation

			after 9/1/2004 should be run to determine if there is unpaid OID and that should be passed here. On all successive calls the Unpaid OID should be the Unpaid OID returned from the previous call.
7.2	Array of Double [61]	Pay	Prospective scheduled payment amounts. This is the scheduled payment amounts as of the date of the last payment. E.g., if you have 179 payments of \$200 and 1 final payment of \$199.95 this field would contain 200 in the first Pay field and 199.95 in the second Pay field. <i>When computing the payment streams, these fields will be set to 0.</i>
7.2 or 4.5	Array of Double [61]	Rate	Prospective schedule of nominal annual interest rates. <i>This field is used for payment stream recalculations only.</i> This will be the current rate at the time of the payment. E.g., if you have a prepayment rate of 4% and a final rate of 10%, the first Rate would contain 4 and the second Rate field would contain 10.***
5.4	Array of Double [20]	Adj	Principal adjustment. <i>This field is used for payment stream recalculations only.</i> Positive values increase the principal, and negative values decrease it. Values between -1 and 1 are considered percentage adjustments (e.g., -0.015 is a 1.5% reduction in principal), while values outside that range are fixed dollar amount adjustments (e.g., 50.00 is a \$50.00 increase in principal). <i>Not yet implemented; set all entries to 0.</i>
3	Array of Integer [61]	PayNP	Prospective number of payments. This is the number of scheduled payments as of the date of the last payment. E.g., if you have 179 payments of \$200 and 1 final payment of \$199.95, you should put 179 in the first PayNP field and 1 in the second PayNP field. This should reflect the number of payments left. <i>This is also used to compute the payment streams. This should be the remaining loan term.</i>
3	Array of Integer [61]	RateNP	Prospective term (number of scheduled payments) of each rate in the Rate array.** <i>This field is used for payment stream recalculations only.</i> E.g., if you have a prepayment rate of 4% and a final rate of 10%, the first RateNP would contain -1 and the second RateNP field would contain 0.
3	Array of Integer [20]	AdjNP	The number of payments to wait until applying the next principal adjustment (Adj). <i>This field is used for payment stream recalculations only.</i> If AdjNP[0] is 0, the first adjustment is applied at the Conversion Date (this could be used for adding a financed conversion fee). Adjustments are the last operation on the principal, taking place after any reduction by the payment. <i>Not yet implemented; set all entries to 0.</i>
2	Integer	LastPayDate Month	Last Payment Made Month. The most recent payment (or other event) date prior to this payment date; it will almost always be "ThisPayDate" from the previous call. If there was no previous payment, the origination date would probably be used.
2	Integer	LastPayDate Day	Last Payment Made Day.

Carleton IRS Student Loan Reporting Module Implementation

4	Integer	LastPayDate Year	Last Payment Made year.
2	Integer	ThisPayDate Month	Actual Payment Date Month. Date payment was actually made. Must be on or after LastPayDate.
2	Integer	ThisPayDate Day	Actual Payment Date Day.
4	Integer	ThisPayDate Year	Actual Payment Date Year.
2	Integer	DueDate Month	Date Payment Due Month. Date payment was actually due. This date must be after LastPayDate.
2	Integer	DueDate Day	Date Payment Due Day.
4	Integer	DueDate Year	Date Payment Due Year.
2	Integer	Conversion Date Month	Date to recognize conversion event(s) (i.e., effect of deferrals, cap interest) Month. <i>This date is used for payment stream recalculations only.</i> This date must be on or after LastPayDate, and on or before DueDate. If the account has not yet converted to repayment, this field will contain conversion to repayment date (probably one month before DueDate). If recalculated during repayment, this should probably be set to zero.
2	Integer	ConversionDate Day	Date to recognize conversion event(s) Day.
4	Integer	ConversionDate Year	Date to recognize conversion event(s) Year.
2	Integer	Interest Handling	Method for interest handling: 0 = U.S. Rule (Escrow interest) 1 = Monthly Actuarial - Compound interest every month 2 = Escrow interest until Conversion Date, then compound 3 = Government pays interest until conversion, then U.S. Rule 4 = Government pays interest until conversion, then compound 5 = Escrow interest until Conversion Date, then compound that, then U.S. Rule thereafter <i>This field is used for payment stream recalculations only. This field will probably contain a 2 or 5 in most cases.</i> <i>Values 3 & 4 are made obsolete by allowing a different rate (zero) before conversion.</i>

**RateNP has two special values. When in interim (prepayment) period, if the first RateNP entry is -1, the first Rate will be used until the conversion date, and then the second Rate will be used. (After the first entry, no entry may be negative.) If any RateNP entry is 0 (zero), the corresponding Rate will be used for the remaining duration of the loan.

***Rate has two possible formats for AS400, either XXXXXX.XX (like the other fixed point fields) or XXXX.XXXXX (five places of precision). To maintain backward compatibility, rates of 100% or more must use the higher precision.

Carleton IRS Student Loan Reporting Module Implementation

OUTPUTREC DEFINITION

Field Length	Field Type	Field Name	Description
7.2	Double	AIP	Updated Adjusted Issue Price
7.2	Double	Yield - Paid	Paid Yield (Original Issue Discount, if QSI is input)
7.2	Double	Yield - Unpaid	Unpaid Yield (Original Issue Discount, if QSI is input)
5.4	Double	Yield Rate	(For auditing purposes only) The annualized yield rate used to compute the earned yield
7.2	Double	Payment	If the payment was to be computed (Pay[0] is 0 in the input), this is the computed regular payment
3	Integer	Term	If the payment schedule was to be computed or adjusted (Principal is greater than 0 in the input), this is the total term
2	Integer	Error Code	An error code may be returned if input values are missing from the input structure or an error occurs during calculation

ERROR CODES/DESCRIPTIONS

Error Code	Description	Troubleshooting
0	No error	
1	Invalid Last Payment Date	Make sure month, day, and year are valid for last payment date.
2	Invalid This Payment Date	Make sure month, day, and year are valid for this payment date.
3	Invalid Next Payment Date	If payment stream is entered, next payment date must be valid.
4	Invalid AIP	AIP needs to be greater than zero.
5	Invalid Number of Payments	Number of payments must be greater than or equal to zero.
6	Invalid Rate Number of Payments	All rate terms after first period must not be negative.
7	Invalid Conversion Date	If principal is greater than zero, make sure month, day, and year of conversion date are valid (or zero, if the conversion date is before LastPayDate).
8	Invalid Payment Streams	If there is AIP remaining, make sure there are payment streams or the means to compute them (input principal).
9	Invalid Adjustment Number of Payments	As long as there are non-zero Adjustment entries, the corresponding terms must not be negative.
10	Interest Short	Payment is not enough to cover interest (when recalculating term with input payment amount).
98	Yield Error	Error solving for Yield rate. Check inputs.
99	Solve Error	Error solving for OID. Check inputs.

Carleton IRS Student Loan Reporting Module Implementation

An error is usually the result of an incorrect input: A required date not filled in, for example. Examine the input, fix the problem, and retry.

If you receive an Interest Short error when recalculating streams, either increase the payments in the schedule, or set the payments to zero and let the module find the payment for a given term.

If you receive a Yield error, make sure the AIP is less than the total payments in the streams. You may need to ask the module to recalculate the streams (making sure the Principal is consistent with the AIP).

If you receive a Solve error with reasonable inputs, or a Yield error after checking the inputs as above, please contact Carleton with the exact inputs that raised the error.

Carleton IRS Student Loan Reporting Module Implementation

EXAMPLES

09/01/03 Disbursement \$4,850.00, with origination fee \$97.00 (AIP \$4,753.00).
 06/05/05 "Conversion" with capitalization of \$355.67 interest at 4%; 5% interest starts.
 07/05/05 Payments scheduled to begin 179 @ \$41.17, 1 @ \$40.16.

Actual Payments Made

07/03/05 - \$41.17
 08/17/05 - \$41.17 (\$15 late fee assessed; \$26.17 P&I)
 09/12/05 - \$50.00
 09/30/05 - \$50.00
 11/01/05 - \$50.00
 12/09/05 - \$40.00

Option 1: Call DLL with payments, no stream adjustments.											
	LastPay Date	ThisPay Date	Due Date	This Pay	Pay [0]	Pay NP [0]	Pay [1]	Pay NP [1]	OID Paid	OID Unpaid	AIP
Input	09/01/03	07/03/05	07/05/05	41.17	41.17	179	40.16	1	*	0**	4753.00
Output									41.17	419.13	5172.13
Input	07/03/05	08/17/05	08/05/05	26.17	41.17	178	40.16	1	*	419.13	5172.13
Output									26.17	424.84	5177.84
Input	08/17/05	09/12/05	09/05/05	50.00	41.17	177	40.16	1	*	424.84	5177.84
Output									50.00	393.50	5146.50
Input	09/12/05	09/30/05	10/05/05	50.00	41.17	176	40.16	1	*	393.50	5146.50
Output									50.00	356.41	5109.41
Input	09/30/05	11/01/05	11/05/05	50.00	41.17	175	40.16	1	*	356.41	5109.41
Output									50.00	328.60	5081.60
Input	11/01/05	12/09/05	12/05/05	40.00	41.17	174	40.16	1	*	328.60	5081.60
Output									40.00	315.82	5068.82
Total Paid OID									257.34		

* OID Paid is an output value.

** OID Unpaid is always zero in the initial input.

***All other input parameters are set to 0. In particular, QSI is set to zero so the "OID" output is actually total yield (paid and unpaid). Carleton recommends this since box 1 on 1098-E is supposed to be total yield (unless you require separate OID for accounting purposes).

