

LoanMaker SP Engine Technical Manual

The LoanMaker SP Engine is a version of the LoanMaker SP Installment Loan Program which is designed to be called by a host program. The Engine can be interfaced with nearly any host program because it uses a reliable and simple system to interchange information. Users can design their own input and output screens to give the host product the desired look and feel.

Input data is transferred to the Engine in a flat ASCII file with "tags" identifying each of the input data items. The Engine is invoked as a spawned process underneath the host program. The host program passes both the name of the file containing the input data and the name of the file to which the Engine will write the results.

The results of the computations or, in the case of an error, the error number and message are returned to the host program in the specified output file. (See Output File Format below.) No messages or data are printed to the screen unless the Engine cannot open the specified output-file or configuration file, or is invoked with an insufficient number of arguments, i.e., no input and output file were specified.

If the specified output file cannot be opened for write access or the configuration file cannot be opened, the appropriate error message is directed to the file "stdout" and the program returns a non-zero value to the host process.

If the specified output file can be opened for write access, the computed results or error message are printed to the file. This file can be a disc file, device, or the console. If an error occurred, the Engine returns a non-zero value to the host process. If no error was detected, it returns a 0.

LMSP Engine Command-Line Options

The program is invoked with the following command:

SP_ENG [-option [-option [...]]] infname outfname

infname is the name of the file containing the data being input to the Engine (default path is the current directory),

outfname is the name of the file the Engine will write the results to (default path is the current directory).

Options are case-sensitive:

-D	Displays a list of the input values captured from the input file. Used to check the data being sent to the Engine by the host program
-M	Prevents the computation of multiple payments with all insurance options for Installment Loans and Single-Payment Loans. Only the specified insurance combination is computed. See Alternate Payment Amounts for Insurance Options below.

-R	Displays a list of the available dealer reserve calculation methods.
-T	Displays a list of the legal tags that can be included in an input file.
-V	Displays a list of the version numbers for the various computational modules.
-cname	Specifies configuration file 'name'. A full path can be specified. The default configuration file is LMSP.DAT located in the same directory as SP_ENG.EXE
-dST	Sets the default plan to ST, e.g., -dCA would select the "CA" insurance plan. See Setting/Changing the Default Plan.
-e	Displays a list of all error and warning numbers and messages.
-h -?	Displays this help screen. (This screen is also displayed if there is an insufficient number of command-line arguments.)
-iname	Displays insurance plan 'name', e.g., -iNM displays the plan named "NM", and -iALL displays all plans available.
-n	Changes the installment-loan first-period interest calendar, i.e., if the configuration file being used specifies an actual calendar, setting this option will change it to a "Federal" calendar, and vice versa.
-r	Changes the installment-loan A.P.R. calendar, i.e., if the configuration file being used specifies a "Federal" calendar, setting this option will change it to an Actual calendar, and vice versa.
-u	This option removes the check on the maximum number of days to the first payment for installment loans. With this option, the Engine will allow any number of days to the first payment.
-y	Shows the <u>annual</u> cost of installment-loan credit insurance (default is the <u>daily</u> cost). If the configuration file already shows the insurance cost on an annual basis, setting this option will show the daily cost. This option only affects the printed loan disclosure. See Printing a Loan Disclosure and Amortization Schedule. (The output file generated by the Engine shows both the daily and annual costs.)

Redirecting Output

Many of the command-line options that send information to the screen send more than one screen and it is usually impossible to read all the information. You can cause the information to be displayed a screen at a time by piping the output to the DOS MORE program, e.g.,

SP_ENG -T | MORE

or you can cause the information to be printed or sent to a file by using the DOS ">" redirector operator, e.g.,

SP_ENG -D IN.TXT OUT.TXT > E:TMP

sends all the values of the input tags found in the input file IN.TXT to the file E:TMP.

Note on Executable File Names and Specifications

The program is named SP_ENG.EXE and will run on an Intel 386 or higher chip under MS-DOS 3.3 or higher. It can be run as a DOS session under Windows (3.1, 95, 98, & NT) and OS/2.

In addition, the credit insurance plan(s) are contained in a configuration file named LMSP.DAT. This file is required and should be located in the same directory as SP_ENG.EXE unless you specify another path and/or name with command-line option "-c".

The program requires about 192K of memory to run. It does not use extended or expanded memory.

Method of Operation

The Engine is designed to be run by a host process and normally does not print any information to the screen. When the Engine is invoked by the host program, the names of two files are specified on the command line. The first file is the name of an Input file which contains the input data and the type of calculation desired. The second file is the name of the Output file to which the Engine will write the computed results.

The Engine must have write access to the drive where the Output file is located.

Typically, the input and output files are quite small and can be located on a RAM drive which will minimize the time needed to read and write the files.

Note that the default directory for the Input and Output files is the current directory (not the directory where SP_ENG.EXE is located). It is generally good practice to specify the full path for both the Input and Output file.

If the Engine is located on a server, you must make sure that the output file names are unique for each process. One way to do this is to have the host program create an empty Output file with a unique name before you call the Engine. The Engine will overwrite the contents of the specified Output file. As an added precaution, you should use the "control_no" tag to include a unique identification string in the Input file. This string is returned in line 6 of the Output file and can be checked to make sure that the proper output file is being read.

Blank lines can appear anywhere in the file and are discarded.

Comments can appear in the file on a line by themselves if preceded with an apostrophe, e.g., the line

' File generated 5/1/95 by Sam Johnson

is treated as a comment and is discarded by the Engine.

A list of legal "tags" can be generated by using the "-T" option.

Input File Date Formats

Dates may be passed to the Engine in any of six formats:

mmddy	013195 or 13195
mmddyyy	01311995 or 1311995
mm/dd/yy	01/31/95 or 1/31/95
mm/dd/yyyy	01/31/1995 or 1/31/1995
yyyymmdd	19950131
yyyy/mm/dd	1995/01/31

The leading "0" for the month is not required for the first 4 formats, however, if the number of days is less than 10, the leading zero **is** required for the number of days, e.g., 10895 and 1/08/95 is okay, but 1895 or 1/8/95 are not okay.

For those formats that use a two-digit year, any year value from 80 through 99 is assumed to be 1980 through 1999, and any year value from 00 through 79 is assumed to be 2000 through 2079.

Specifying the Loan Type

The tag "loan_type" is used to specify what type of loan calculation the Engine is to perform. A loan_type of "0" specifies an installment loan which is the default. If no loan_type is specified, an installment loan will be computed.

Make sure to specify the loan type if you are calculating other than an installment loan (it's recommended that you always set the loan type). If the loan type is not specified, the Engine will assume an installment loan is being calculated and may return seemingly nonsensical errors if some other calculation is expected.

loan_type	<p>A value of 0 calculates an installment loan. This is the default and if the loan_type is not specified, the Engine will attempt to compute an installment loan using the data in the Input file.</p> <p>A value of 1 calculates a single-payment loan.</p> <p>A value of 2 calculates a revolving credit loan.</p>
------------------	---

	<p>A value of 3 performs an A.P.R. Check & Dealer Reserve calculation.</p> <p>A value of 4 executes the qualification routine.</p>
--	--

General Output File Format

Regardless of the type of calculation requested, the first six lines of the output file created (or overwritten) by the Engine are the same.

The first three lines contain the program banner information identifying the Engine being used.

The fourth line contains either an ASCII "0" if no error occurred, or the error or warning number in ASCII, e.g., "200". (The quotes do not appear in the Output file.)

The fifth line contains the error or warning message or is blank (carriage return/line feed) if no error or warning occurred.

The sixth line contains the control number read from the input file. Only the first contiguous segment is returned if there are any embedded blanks or tabs in the passed control number.

If no error occurred, then the remainder of the output will appear with tags appropriate for the calculation being performed. See the individual sections in this manual.

Installment Loans and Single-Payment Loans

To calculate an installment loan, the "loan_type" can be passed as 0 in the input file. (It can be omitted for installment loans because the "loan_type" defaults to 0.) To calculate a single-payment loan, the "loan_type" must be passed as 1. The other "tags" that need to be passed are described below (those marked with an asterisk are required).

Input File Tags

loan_type*	<p>This tag is used to tell the Engine what type of loan is being calculated.</p> <p>A value of 0, which is the default, calculates an installment loan.</p> <p>A value of 1 calculates a single-payment loan.</p>
dis_code	The disability code (0 for none, 1 for plan 1, 2 for plan 2, 3 for plan 3, or 4 for plan 4). Disability insurance is only available on installment loans.
dis_term	The term of disability insurance for installment loans. Only required if truncated insurance is desired. Disability term defaults to the loan term if not specified.
fees	The amount of fees paid to others on behalf of the borrower. Included in the amount financed. Not to be used for prepaid charges.

first_date *	The date the first payment is due. Must be after the note_date and loan_date. Can be passed in any of several date formats. See Date Formats. To compute a single-payment loan, pass the payment due date in this field.
int_rate *	The loan interest rate. Can be passed with any number of decimal places.
iui_code	The involuntary unemployment insurance code (0 = none, 1 = single).
iui_term	The term of involuntary unemployment insurance.
life_code	The life insurance code (0 = none, 1 = single, 2 = joint). Used for both installment and single-payment loans.
life_term	The term of life insurance (required only if less than the loan term). Defaults to the loan term.
loan_ant	The initial amount of the loan. If there are any prepaid charges, they are deducted from this amount to determine the proceeds to the borrower.
loan_date *	The date the finance charge begins to accrue. Generally the same as the note date, but it can be later.
loan_term *	The term of the loan in months. The term is ALWAYS passed in months even for non-monthly loans. To compute a single-payment loan, pass a term of 0 months.
note_date *	The date the loan is made.
pins_code	The property insurance code (0 = none, 1 = yes).
pins_term	The property insurance term (required only if less than the loan term).
pnt_freq	The number of payments to be made each year, e.g., 52 for weekly payments, 26 for biweekly, etc. Defaults to 12 for monthly payments. The value in this field has no meaning for single-payment loans.
ppd_chg	The total of prepaid finance charges. This amount is deducted from the loan amount to determine the proceeds to the borrower. To finance the prepaid charges, increase the loan amount by the amount of the prepaid charges and also include them in this field.
spec_1	The first irregular payment specifier. This must be one of the legal payment specifiers. See Installment Loan Irregular Payment Specifiers.
spec_1_ant	The amount associated with the first irregular payment specifier. If not passed, this value defaults to 0, which indicates that the payments referenced by irregular payment specifier "spec_1" are to be skipped (i.e., have a value of 0).
spec_2	The second irregular payment specifier.
spec_2_ant	The amount associated with the second irregular payment specifier.
spec_3	The third irregular payment specifier.

spec_3_amt	The amount associated with the third irregular payment specifier.
spec_4	The fourth irregular payment specifier.
spec_4_amt	The amount associated with the fourth irregular payment specifier.
spec_5	The fifth irregular payment specifier.
spec_5_amt	The amount associated with the fifth irregular payment specifier.
state_abbr	The name of a state or insurance plan in the configuration file. This entry controls the credit insurance plan as well as some other configuration options. See <u>Setting/Changing the Default Plan</u> .
control_no	A value that is returned by the Engine in the specified output file. The calling program should put some unique text string in this field and then check that the same string is returned in the output file as a method of ensuring that the correct output file has been retrieved. If there are any embedded spaces or tabs within the control number, only the first segment will be returned.

Setting/Changing the Default Plan

If the insurance plan (usually a state abbreviation, e.g., "NY") is specified in the Input file with the "state_abbr" tag, that plan is used if it exists. If the plan does not exist, an error is reported.

It is not necessary to specify a plan in the Input file, and if none is specified, the Engine will use the default plan. The default plan is specified in the configuration file and can be determined by viewing the file with the "-i" option.

You can override the default plan specified in the configuration file by using the "-d" command-line option. The plan specified must exist in the configuration file or an error will occur.

Specifying Irregular Payments in Installment-Loans

LoanMaker SP allows you to specify irregular payments during an installment loan. These can include skipped payments, pickup payments, balloon payments, repeated payments, and regular payment amounts, all of which can be specified in any combination.

Irregular payments are specified using the tags "spec_x" and "spec_x_amt" where "x" is a number from 1 to 5. The first ("spec_x") is for the "specifier", and the second ("spec_x_amt") for the amount to be assigned to those payments identified by the specifier. (The amount is zero for skipped payments.)

The specifiers can any of the following:

mm	representing a month number, e.g. 05.
myy	representing a month and year, e.g., 1295.
yyy	representing a year, e.g., 1995
@	representing all payments except the final payment
F	representing the final payment.

R##	representing a series of repeated payments.
M##	to move the repeat payment pointer.

Payment specifiers can be used in any order, however, the order in which they are assigned is significant in determining the results. Often, a payment can be specified more than once and it is the last specifier on the list that has effect.

For example, if we want to set all the payments to \$100 and solve for the balloon payment, we would use the '@' specifier, e.g.,

```
spec_1           @
spec_1_ant       100.00
```

Now if we want to skip the payments that fall in June, we will use the '06' specifier with a payment amount of \$0.00, and we have to use it after the '@' specifier, e.g.,

```
spec_1           @
spec_1_ant       100.00
spec_2           06
spec_2_ant       0.00
```

(If we had assigned the '06' specifier to spec_1 and the '@' to spec_2, the '@' would override the skipped specification and the June payments would not be skipped.)

An example of a complex transaction is as follows: Assume we want a loan with a final balloon payment of \$20,000, the payments in June and July to be skipped, and all December payments to be pickup payments in the amount of \$2000.00. Compute the regular monthly payment.

The specifiers would appear as follows:

```
spec_1           F
spec_1_ant       20000.00
spec_2           06
spec_2_ant       0.00
spec_3           07
spec_3_ant       0.00
spec_4           12
spec_4_ant       2000.00
```

As another example, assume we want to calculate the final balloon payment of a loan with all payments fixed at \$500 except for the June payments which are \$800, and the January, 1995 and January 1997 payments which are skipped. The specifiers would appear as follows:

```
spec_1           @
spec_1_ant       500.00
spec_2           06
spec_2_ant       800.00
spec_3           0195
spec_3_ant       0.00
spec_4           0197
spec_4_ant       0.00
```

(Note that the "@" specifier usually has to be assigned to spec_1, as it is in this example. If it were not, it would override the other specifications and we would have a balloon loan with all payments set to \$500.00.)

By using the "yyyy" specifier, we can affect all the payments that fall in a particular year. The following specifiers would set all payments to \$500 except that all payments in 1996 would be set to \$700 and the December payments would be skipped. The final payment is to be computed.

```

spec_1           @
spec_1_ant       500. 00
spec_2           1996
spec_2_ant       700. 00
spec_3           12
spec_3_ant       0. 00
  
```

If you specify a payment that does not fall within the loan term, the specifier has no effect. For example, a month can be specified that is either before the loan begins or after the loan has matured.

For quarterly, semi-annual and annual loans, no payment may be due in a particular month that is specified. An example of this is a quarterly loan with the first payment due in January. Since payments are due only in the months of January, April, July and October, payment specifiers for any other months are ignored.

Specifying a Repeating Series of Payments

The 'R' specifier is used to set a series of payments to a certain value (which can be 0 indicating the series of payments is to be skipped [the first or last payment in a loan cannot be skipped]). For example, to set the first 12 payments of a loan to \$100.00, issue the following specifier:

```

spec_1           R12
spec_1_ant       100. 00
  
```

You can have more than one series, for example:

```

spec_1           R12
spec_1_ant       100. 00
spec_2           R6
spec_2_ant       150. 00
spec_3           R12
spec_3_ant       175. 00
  
```

would set the first 12 payments to \$100.00, the next six to \$150.00, and the 12 after that to \$175.00. This loan would have to have at least 31 payments because we have set the first 30 payments to specific values.

To specify a series of repeat payments that does not start at the first payment or is non-continuous, you can use the 'M' specifier to move the repeat payment pointer over one or more payments without assigning them a value. For example, assume we want to set the last 36 payments of a 48-month loan to \$100.00 and compute the first 12:

```

spec_1           M12           ' move past the first 12 payments
spec_1_ant       0. 00
  
```

spec_2	R36	' set the last 36 to \$100.00
spec_2_ant	100.00	

Similarly, if we want to set the first and last 12 payments at \$200.00 on a 48-month loan and compute the series of 24 payments in the middle:

spec_1	R12	' set the first 12 to \$200
spec_1_ant	200.00	
spec_2	M24	' move past the middle 24 payments
spec_2_ant	0.00	
spec_3	R12	' set the next (last) 12 to \$200.00
spec_3_ant	200.00	

Any series of payments that does not include the first payment can be skipped by setting the payment amount to 0 (the first payment cannot be skipped).

Repeat payment specifiers can be mixed with other specifiers and follow the same rules of precedence.

As an example, to specify the first 12 payments at \$100.00, the second 12 at \$200.00, and set the July payments to \$4.00, use the following:

spec_1	R12	' set first 12 to \$100.00
spec_1_ant	100.00	
spec_2	R12	' set next 12 to \$200.00
spec_2_ant	200.00	
spec_3	07	' set all July payments to \$4.00
spec_3_ant	4.00	

The '07' specifier takes precedence over the repeat payment specifiers and will change all July payments to \$4.00.

Installment-Balloon Loans

Balloon loans can be solved using three general methods: First, we can specify the regular payment amount and compute the final payment. Second, we can specify the final payment amount and compute the regular payment. Third, we can compute the regular payment amount based on a fully-amortizing term and then use that computed payment amount to compute the balloon due at the end of some shorter period. For example, we can calculate the payment based on a 15-year amortization and then use that payment to calculate a balloon due in 5 years.

Installment Loans with Other-than-Monthly Payment Frequencies

The irregular payment specifiers work the same way for other-than-monthly loans. Some, such as the 'mm' and 'mmyy' affect all payments that fall within the specified month.

The repeat payment specifiers 'R' and 'M' affect payments by their number within the loan (not by the month in which they fall).

For weekly and biweekly loans, all payments that fall within the specified month or months are affected, thus the specifier

spec_1	05
spec_1_ant	50.00

would set all payments that fall within May to \$50.00. For weekly payments, this would usually be 4 payments, but could be 5 because there are 13 weeks in three months, and a weekly due date will occur 5 times in a month once every three months. Similarly, for biweekly payments, usually 2 payments would be affected, but once every six months, it would be 3 payments.

For quarterly, semiannual, and annual payment frequencies, the payment(s) that fall within the specified month(s) are affected. If no payment falls within the specified month(s), the specifier is ignored. (This can happen with quarterly, semiannual, and annual payment frequencies.)

As an example of an irregular loan with semiannual payments, we could specify \$4000 semiannual payments for 1995, \$3500 payments for 1996, and a balloon payment in 1997, which is to be calculated. Assuming a loan date of 1/1/94, the term of the loan would be 30 months (2-1/2 years). The specifiers would be

spec_1	1995	set 1/1/95 and 7/1/95 pmts
spec_1_ant	4000.00	
spec_2	1996	set 1/1/96 and 7/1/96 pmts
spec_2_ant	3500.00	

The final payment, due on 1/1/1997, is calculated.

Printing a Loan Disclosure and Amortization Schedule

The Engine will print an amortization schedule and/or disclosure.

To request a schedule, include in the Input file the tag "prt_am_sched" and specify the name of the file to which you want the schedule sent, e.g., the line

prt_am_sched d:amsched.txt

in the input file will cause the Engine to write the current loan amortization schedule to the file "d:amsched.txt". The file can be a disc file or a device, e.g., use "LPT1" if you want to print the amortization schedule on a printer attached to LPT1.

Similarly, to request a loan disclosure, include in the Input file the tag "prt_disc" and specify the name of the file to which you want the disclosure sent, e.g., the line

prt_disc lpt2

will cause a loan disclosure to be printed on the printer attached to LPT2.

Alternate Payment Amounts for Insurance Options

For installment and single-payment loan calculations, the Engine normally computes alternate payment amounts for all insurance options and returns the values of the computed payment and the total insurance premium for each option. (For single-payment loans, only the alternate life insurance options are calculated because no disability insurance is available on single-payment loans.)

If the user did not specify disability insurance, the first disability plan is used for the alternate payment calculations with disability insurance. If the user selected a disability plan, the selected plan (1, 2, 3, or 4) is used for the alternate payment calculations.

For revolving credit calculations, the Engine normally computes the monthly insurance premium for all types of insurance.

If the `"-M"` option is specified, these alternate payments are not computed.

Because of possible underwriting limits, it is not always possible to compute a payment amount for all combinations of insurance. For any particular insurance combination that is not available, the computed payment and total premium amount are assigned a value of 0.

(An example of this is when a user requests alternate payments for a 180-month loan and the maximum disability term allowed is 120 months. In this case, the payment and total premium amounts for insurance combinations that include disability will be returned as 0. Another example is that there is no disability plan installed for a state or the user has selected a nonexistent plan [e.g., the user selected disability plan 3 and there is only 1 plan available in the particular state].)

The alternate payment amount returned is the "computed payment". For regular installment loans, this is the payment the borrower will make. For balloon loans, the computed payment could be either the periodic payment or the final payment, depending upon which the Engine was asked to compute. For single-payment loans, this is the single payment due at maturity.

Similarly, for irregular loans, the computed payment is the unknown payment that the Engine was asked to compute.

Installment Loan and Single Payment Loan Output File Format

The first six lines in the output file are standard, i.e., 3 lines for the program banner, an error number or 0 if no error occurred, the error message or blank if no error occurred, and, on line 6, the control number.

If no error occurred, the alternate payment and total premium amounts appear next unless they have been suppressed with the `"-M"` option. These items always appear in the same order:

pnt_sl	Payment with single life only.
prem_sl	Premium with single life only.
pnt_slh	Payment with single life and disability.
prem_slh	Total premium with single life and disability.
pnt_jl	Payment with joint life only.
prem_jl	Premium with joint life only.
pnt_jlh	Payment with joint life and disability.
prem_jlh	Total premium with joint life and disability.
pnt_h	Payment with disability only.
prem_h	Premium with disability only.
pnt_ni	Payment with no insurance.

If a particular insurance combination is not available, a value of 0 is returned for both the payment and premium amount. Following the alternate payment options, the loan disclosure appears with the requested insurance options:

apr	The annual percentage rate.
fin_chg	The finance charge.
ant_fin	The amount financed.
tot_pnts	The total of all periodic payments.
fees	The amount of financed fees paid to others (filing fees, etc.).
ppd_chg	The prepaid finance charge amount.
note_ant	The initial principal amount of the loan.
int_rate	The loan interest rate.
life_prem	The credit life premium. If no life was requested, this will be 0.
life_ben	The initial amount of life insurance.
life_term	The term of life insurance (in months) or 0 for single-payment loans.
life_cost_day	The cost of life insurance per day returned as a string, e.g., "Life cost is \$0.19/day".
life_cost_yr	The cost of life insurance per year returned as string, e.g., "Life cost is \$ 68.01/yr". Available only for installment loans.
dis_plan	The disability plan used to compute the disability premium, e.g., "ID14R" or, for single-payment loans, the state abbreviation used for life premium calculations.

The disability fields are all returned as 0 for a single-payment loan.

dis_prem	The disability premium. If no disability insurance was requested, this will be 0.
dis_ben	The monthly disability benefit.
dis_term	The term of the disability coverage (in months).
dis_cost_day	The cost of disability insurance per day returned as a string, e.g., "Disability cost is \$0.80/day".
dis_cost_yr	The cost of disability insurance per year returned as a string, e.g., "Disability cost is \$ 290.39/yr".
iu_i_prem	The involuntary unemployment insurance premium.
iu_i_ben	The involuntary unemployment insurance benefit per month.
iu_i_term	The involuntary unemployment insurance term.
pins_prem	The property insurance premium.
pins_ben	The amount of property insurance.
pins_term	The property insurance term
loan_term	The term of the loan (in months) or, if a single-payment loan, 0.
tot_no_pnts	The total number of periodic payments.
freq_name	The frequency of periodic payments returned as a string, e.g., "Monthly", or for single-payment loans, the string "Single payment".

Next, the payment schedule is printed. The first line contains the tag "pmt_sched" with no value following it. This is followed by 1 or more lines containing the payment schedule. At the end of the payment schedule, a line containing a single "0" appears, e.g.,

```
pmt_sched
12      1099.97  10/01/95
0
```

which indicates that there are 12 payments of \$1099.97 starting on 10/01/95. To indicate the end of the payment schedule, a line is printed with a single "0".

For loans with more than one payment level, a line is printed for each level, e.g., a simple balloon loan would have a schedule similar to the following:

```
pmt_sched
35      455.12   10/01/95
1      4000.00   9/01/98
0
```

As another example, this is a 36-month balloon loan with the regular payment fixed at \$125.00 and a pickup payment in the amount of \$750 due each May. The final balloon payment was calculated:

```
pmt_sched
7      125.00   10/01/93
1      750.00   5/01/94
11     125.00   6/01/94
1      750.00   5/01/95
11     125.00   6/01/95
1      750.00   5/01/96
3      125.00   6/01/96
1     11323.83   9/01/96
0
```

Sample Output File for an Installment Loan

LoanMaker SP Engine - Version 1.xx

Licensed to:

Copyright 1995-99 PROM Software, Burlington VT USA

0

control_no R5565-900

apr 12.249
fin_chg 10414.35
amt_fin 14087.45
tot_pmts 24501.80
fees 5.55
ppd_chg 0.00
note_amt 14087.45
int_rate 12.250
life_prem 410.94
life_ben 14087.45
life_term 120
life_cost_day Life cost is \$0.11/day
life_cost_yr Life cost is \$ 41.09/yr
dis_plan NY14R
dis_prem 1670.96
dis_ben 204.18
dis_term 120
dis_cost_day Disability cost is \$0.46/day
dis_cost_yr Disability cost is \$ 167.10/yr
iui_prem 0
iui_ben 0
iui_term 0
pins_prem 0
pins_ben 0
pins_term 0
loan_term 120
tot_no_pmts 120
freq_name Monthly
pmt_sched
5 211.38 10/01/93
1 125.00 3/01/94
11 211.38 4/01/94
1 125.00 3/01/95
11 211.38 4/01/95
1 125.00 3/01/96
11 211.38 4/01/96
1 125.00 3/01/97
11 211.38 4/01/97
1 125.00 3/01/98
11 211.38 4/01/98
1 125.00 3/01/99
11 211.38 4/01/99
1 125.00 3/01/00
11 211.38 4/01/00
1 125.00 3/01/01
11 211.38 4/01/01
1 125.00 3/01/02
11 211.38 4/01/02
1 125.00 3/01/03
6 211.38 4/01/03

0

Sample Output File for a Single Payment Loan

LoanMaker SP Engine - Version 1.xx

Licensed to:

Copyright 1995-97 PROM Software, Burlington VT USA

0

control_no AB34XC-8

pnt_sl 12791.36
prem_sl 53.28
pnt_slh 0.00
prem_slh 0.00
pnt_jl 12827.96
prem_jl 87.79
pnt_jlh 0.00
prem_jlh 0.00
pnt_h 0.00
prem_h 0.00
pnt_ni 12734.85

apr 13.145
fin_chg 779.30
ant_fin 11955.55
tot_pnts 12734.85
fees 5.55
ppd_chg 50.00

note_ant 12005.55
int_rate 12.250

life_prem 0.00
life_ben 0.00
life_term 0
life_cost_day
life_cost_yr

dis_plan OR
dis_prem 0.00
dis_ben 0.00
dis_term 0
dis_cost_day
dis_cost_yr

loan_term 0
tot_no_pnts 1
freq_name single payment

pnt_sched
1 12734.85 3/01/94
0

Revolving Credit Loans

The Engine can be used to compute the minimum monthly payment and various insurance options for an open-end revolving credit loan. Typically, this routine is used to illustrate to a borrower what the minimum payment and credit-insurance premiums would be for a certain outstanding balance.

Revolving Credit Loans Input File Tags

Items marked with an asterisk are required.

rc_bal *	The balance or credit line amount to be used to compute the monthly payment and premiums. Used for revolving credit loans only. This amount must be specified to obtain payment and premium computations.
rc_pct_bal *	The percentage applied to the balance to determine the monthly payment. Used for revolving credit loans only. This amount must be specified to obtain payment and premium computations.
rc_pmt_min	The minimum monthly payment amount. Used for revolving credit loans only. Default is 0 which indicates no minimum payment.
rc_days	The number of days in the billing cycle (affects the premium calculations). Default is 30 days.
rc_life_code	The life insurance code (0 = none, 1 = single, 2 = joint). Used for revolving credit loans only. Default is 0.
rc_dis_code	The disability code (0 = none, 1 = single, 2 = joint, if available). Used for revolving credit loans only. Default is 0.
rc_iui_code	The involuntary unemployment insurance code (0 = none, 1 = single). Used for revolving credit loans only. Default is 0.

Revolving Credit Output File Format

The first six lines in the output file are standard, i.e., 3 lines for the program banner, an error number or 0 if no error occurred, the error message or blank if no error occurred, and, on line 6, the control number.

The output data appears in the following order with the following tags:

prem_sl	The monthly premium for single life insurance.
prem_jl	The monthly premium for joint life insurance.
prem_h	The monthly premium for single disability insurance.
prem_jh	The monthly premium for joint disability insurance.
prem_iui	The monthly premium for involuntary unemployment insurance.
rc_pnt	The monthly payment due (based on the passed balance).
rc_life_ben	The amount of life insurance for the current month (limited, if appropriate, by the maximum allowable amount).
rc_life_prem	The life premium for the life code specified.
rc_dis_ben	The disability benefit (limited, if appropriate, by the maximum monthly benefit).
rc_dis_prem	The disability premium for the disability code specified.
rc_iui_ben	The involuntary unemployment insurance benefit (limited, if appropriate, by the maximum monthly benefit).
rc_iui_prem	The involuntary unemployment insurance premium for the I.U.I. code specified.

Sample Output File for a Revolving Credit Loan

LoanMaker SP Engine - Version 1.xx

Licensed to:

Copyright 1995- 97 PROM Software, Burlington VT USA

0

control_no 33A34ZC- 8

prem_sl 7. 20

prem_jl 11. 52

prem_h 10. 80

prem_jh 14. 40

prem_iui 12. 49

rc_pnt 360. 00

rc_life_ben 12000. 00

rc_life_prem 7. 20

rc_dis_ben 360. 00

rc_dis_prem 10. 80

rc_iui_ben 360. 00

rc_iui_prem 12. 49

A.P.R. Check & Dealer Reserve Routine

Description

The A.P.R. Check and Dealer Reserve Routine can be used for two functions. The first verifies that the loan has been disclosed properly, and the second computes the dealer reserve and bank retention. This routine is designed to check automobile and other indirect loans where a dealer reserve amount is paid. Loans checked with this routine must be regular installment loans (all payments are the same) or a balloon payment loan (the last payment can be different).

The first function checks a loan to make sure that the items in the "Federal Box" are arithmetically correct and within the specified tolerances. The interest rate and annual percentage rate are computed and checked. Installment loans with an optional balloon payment and a variety of payment frequencies (weekly, biweekly, monthly, quarterly, semiannual, and annual) can be checked.

Several arithmetic checks are performed first: the payment schedule total must equal the total of payments, the amount financed plus finance charge must equal the total of payments, the amount financed plus prepaid charges must equal the note amount, and so forth.

The A.P.R. is checked against the finance charge to make sure it is within the tolerance specified. The tolerance defaults to 0.05% but can be set at any value greater than 0.01%. (This means that if the disclosed A.P.R. is within the specified tolerance, it is considered to be okay.)

The interest rate is then checked against the interest charge to make sure that it is within the tolerance specified. The tolerance defaults to 0.05% but can be set at any value greater than 0.01%.

The interest rate and A.P.R. checks are based on the actual loan date and first payment due date. The user can specify either the "Federal" calendar or an actual calendar to measure the length of the first period.

If the loan being checked fails any of the above tests, an error is reported. If the loan passes all tests and a dealer reserve type has been specified, the dealer reserve and bank retention are calculated. (Loan can be checked without calculating a dealer reserve.)

The dealer reserve can be computed by any of 10 methods (see table below). For any dealer reserve method, a percentage of the calculated reserve can be paid, a minimum reserve amount can be calculated, and, if desired, the dealer reserve can be paid on an amount different than the note amount or amount financed.

Dealer Reserve Tags

The items marked with an asterisk are required. The other items are either optional or will default to a value.

loan_type*	The loan type must be set to a value of 3 to do a dealer reserve calculation.
rs_ant_fin*	The amount financed from the "fed" box on the contract being checked.

rs_ant_pd_on	The amount that the dealer reserve is paid on. Defaults to the note amount if not passed.
rs_apr*	The annual percentage rate from the "fed" box on the contract being checked.
rs_buy_rate	The interest rate that the contract is being "bought" or discounted at. For dealer reserve types 7 and 8, this is the percentage of the interest charge or note amount, respectively. Only required if the dealer reserve is being computed.
rs_fin_chg*	The finance charge from the "fed" box on the contract being checked.
rs_flat_fee	The flat fee being paid to the dealer if the flat fee dealer reserve method is being used. (Only required if the flat-fee dealer reserve method is being used.)
rs_int_rate	The interest rate on the contract -- defaults to the annual percentage rate if not specified.
rs_min_resv	The minimum amount paid to the dealer regardless of the calculated dealer reserve.
rs_note	The initial principal amount of the loan. Defaults to the amount financed if not specified.
rs_pct_pd	The percentage of the total calculated dealer reserve paid. Defaults to 100% if not specified.
rs_pmt*	The regular payment amount.
rs_pmt_f	The balloon payment amount. If not specified, all payments are considered equal to the rs_pmt amount specified above. For purposes of computing the dealer reserve, this should only be specified if the loan is actually a balloon loan. (See discussion under Dealer Reserve Types - 1: Difference in Charges.)
rs_ppd_chg	The prepaid finance charge from the "fed" box on the contract being checked.
rs_tol_rate	The tolerance used to check the interest rate on the contract. Defaults to +/- 0.05% if not specified. Set to a very large value if you do not wish to check the interest rate, e.g., rs_tol_rate 100.
rs_tol_apr	The tolerance used to check the annual percentage rate on the contract. Defaults to +/- 0.05% if not specified. Set to a very large value if you do not wish to check the annual percentage rate.
rs_tot_pmts*	The total of payments from the "fed" box on the contract being checked.
rs_cal_apr	The calendar used to measure the length of the first period to check the annual percentage rate. Default is "0" for the "fed" calendar. Specify a "1" for an actual calendar.

rs_cal_int	The calendar used to measure the length of the first period to check the interest rate and to calculate the dealer reserve amount. Default is "0" for the "fed" calendar. Specify a "1" for an actual calendar.
rs_freq	The number of payments per year. Default is 12 (monthly).
rs_no_pnts*	The total number of payments to be made on the contract.
rs_trun_no_pnts	Used for truncated dealer reserve types only. The number of payments that dealer reserve is paid on. Must be less than the total number of payments and can only be specified for truncated dealer reserve types.
rs_resv_type	The dealer reserve type to use. Default is 0 which means do not calculate the dealer reserve.
rs_l_date*	The loan date (date the finance charge begins to accrue).
rs_f_date*	The first payment due date.

A.P.R. Check and Dealer Reserve Output File Format

The first six lines in the output file are standard, i.e., 3 lines for the program banner, an error number or 0 if no error occurred, the error message or blank if no error occurred, and, on line 6, the control number.

The output data appears in the following order with the following tags:

rs_desc	Text description of the dealer reserve method used, e.g., "Difference in charges".
rs_dlr_resv	The dealer portion of the finance charge.
rs_bank_retent	The bank portion of the finance charge.
rs_loan_status	Returns with "OK" if the loan passes all tests, otherwise no argument is returned after rs_loan_status.

Dealer Reserve Errors

If the loan fails one or more of the validation checks, an error is reported. Only the first error detected is reported. (Dealer reserve errors are also reported if any of the input data is illegal, such as an illegal payment frequency.)

If you do not want to check the interest rate, set the interest rate tolerance at a very high value (say 100%). Similarly, if you do not want to check the annual percentage rate, set the A.P.R. tolerance at a very high value.

If the loan passes all validation checks, the loan "status" is returned as "OK". The dealer reserve is only calculated if 1) the loan passes all validation checks, and 2) a dealer reserve type was specified.

Sample Input File for a Dealer Reserve Calculation

loan_type 3
rs_ant_fin 9950
rs_apr 12.52
rs_buy_rate 12.25
rs_fin_chg 2748.88
rs_flat_fee 65
rs_int_rate 12.25
rs_min_resv 25
rs_note 10000
rs_pct_pd 100
rs_pnt 264.56
rs_ppd_chg 50
rs_tol_rate 0.10
rs_tol_apr 0.10
rs_tot_pnts 12698.88
rs_cal_apr 0 ' 0 == fed, 1== act
rs_cal_int 0 ' 0 == fed, 1== act
rs_freq 12
rs_no_pnts 48
rs_resv_type 1 ' difference in charges
rs_l_date 12/01/93
rs_f_date 1/01/94

Sample Output File for Dealer Reserve Calculation

LoanMaker SP Engine - Version 1.xx
Licensed to:
Copyright 1995-97 PROM Software, Burlington VT USA
0

control_no
rs_desc Difference in charges.
rs_dlr_resv 25.00
rs_bank_retent 2723.88
rs_loan_status OK

Description of Dealer Reserve Methods

This program contains several different methods of computing the dealer reserve that are described below.

1: Difference in Charges - The entered simple buy rate is applied (using the simple method of compounding interest) to the note amount over the term of the loan to find the bank's retention. If there is a prepaid charge, it is added to the bank's retention. The bank retention is subtracted from the total finance charge to compute the dealer reserve. This method is commonly used by financing companies of automobile manufacturers and some banks.

A special case exists if you enter a balloon payment that is identical to the regular periodic payment. The balloon payment is "frozen" and the remaining N-1 payments are recomputed to determine the bank's retention. Because the regular periodic payments are now slightly lower than the final payment, the loan amortizes more slowly and develops a slightly higher finance charge. This results in a higher bank retention and lower dealer reserve.

This happens if you set a value to the rs_pmt_f field for a non-balloon loan. Thus, for non-balloon loans, the rs_pmt_f field should not be assigned a value (or it can be assigned a value of 0). This signals the program to use the regular method of computing the reserve.

4: Present Value Reserve - This method discounts the stream of periodic payments and the balloon payment, if any, at the simple buy rate (using the actuarial method of compounding interest) to find their present value. The note amount is subtracted from this present value to find the dealer reserve. The bank's retention is the difference between the contract finance charge and dealer reserve. This method provides the contract buyer with a yield equal to the simple buy rate if the dealer reserve is paid on the loan date.

5: Pro-rate Reserve - This method is designed to compute the total reserve the dealer will receive when the reserve is paid over the life of the loan. In this system, each month the dealer receives a portion of the interest collected by the bank. The total bank's retention is computed by prorating the simple buy rate to the contract interest rate. The dealer reserve is the difference between the contract finance charge and the bank's retention.

6: Flat Fee per Contract - This method pays the dealer a flat fee for the contract.

7: Percentage of Interest Charge - This method pays the dealer a percentage of the contract interest charge (excluding any prepaid charge).

8: Percentage of Note Amount - This method pays the dealer a percentage of the note amount, much like an origination fee on a mortgage.

11: Truncated difference in charges (by amortization) - This method is similar to method 1 above. A hypothetical payment is computed using the buy interest rate. The loan is then amortized using both the contract payment and this hypothetical payment over the truncated reserve term. The dealer reserve is the difference in the sum of the interest charges over the truncated term.

12: Truncated difference in charges (by payment difference) - This method is a simpler method of type 11 above. A hypothetical payment is computed using the buy interest rate. The dealer reserve is then the difference between the hypothetical payment and the contract payment

multiplied by the number of payments in the truncated term. This method is easy to calculate but not as accurate as types 11 and 15.

15: Truncated pro-rate (by amortization) - This method amortizes the loan over the truncated term summing the portion of the interest charge each period that is to be paid to the dealer. The portion is based on the ratio of the buy rate to the contract rate, thus if the buy rate is 8% and the contract rate 10%, the dealer will receive 2/10th of the interest charge each period that is within the truncated term.

16: Truncated pro-rate (by ratio) - This method computes the dealer reserve as a simple ratio of the buy interest rate to the contract rate, and the truncated term to the total term. Thus for a 10% 120-month contract rate with an 8% buy rate for a truncated term of 60 months, the dealer reserve would be 2/10ths x 60/120ths of the total interest charge. This method is easy to calculate but not as accurate as types 11 and 15 above.

Negative Dealer Reserve (Buy-down) Calculations

If the buy rate is the same or higher than the contract interest rate and there is no minimum dealer reserve specified, the dealer reserve amount will become negative.

This negative amount is the amount the dealer should pay the bank to accept the contract to make up for the low interest rate. For example, if a contract is written at 4% but the bank buys the contract at 8%, the dealer would have to pay the bank some additional amount (a "negative" dealer reserve).

If the contract interest rate and buy rate are the same, the dealer reserve may show a small negative amount. This is usually because of the way the payment is rounded on the original contract and represents the pennies lost due to the rounding method. Unless you are attempting to calculate a negative dealer reserve, it is recommended that you set the minimum dealer reserve at some small value less than a cent (say \$0.001) to make sure the dealer reserve amount is never less than zero.

Qualify Routine

This routine computes the available loan amount from the specified payment amount, term, interest rate, and frequency of payments. Use a "loan_type" of 4 to run the qualify routine.

The input tags used are:

qu_int_rate	The interest rate to be used.
qu_pnt	The periodic payment amount.
qu_term	The term (in months) of the loan.
qu_freq	The number of payments made per year (default is 12 for monthly payments).

Sample Qualify Input File

loan_type 4

qu_term 48
qu_int_rate 12.25
qu_pnt 300

Sample Qualify Output File

LoanMaker SP Engine - Version 1.xx

Licensed to:

Copyright 1995-97 PROM Software, Burlington VT USA - All Rights Reserved

0

control_no
qual_loan 11339.26
qual_no_pnts 48

Error and Warning Messages

Several types of error messages and a warning message can be returned.

The first type of error message is returned by the Engine shell. These are numbered from 200 and up and generally occur because the Engine cannot open an input file or output file specified for printing the Installment Loan amortization schedule or disclosure.

The only fatal errors which result in the sending of an error message to “stdout” occur when the Engine is unable to open the specified output file or configuration file. If this happens, the Engine terminates with a non-zero value returned to the calling process and prints the error message to “stdout”. These are the only messages printed to stdout.

If the Engine cannot create the Output file, it is usually because the output file is either locked by another process, write-protected, or the file name is illegal. If the Engine cannot find the configuration file, make sure the default configuration file LMSP.DAT is located in the same directory as the Engine (this may not be the current directory). If you have specified a configuration file, make sure the Engine has read access to it and that the path is correct.

If any other errors are detected, they are placed on the specified output file.

The second type of error message is returned by the LoanMaker SP computation module. There are several of these which generally relate to illegal or missing input values. They are numbered from 1 to 99 (only about 75 are currently used). If one of these errors occurs, its number and the corresponding message are returned in the specified output file. The Engine terminates with a non-zero value.

There is only one warning message (number 100) which is issued if the payment schedule contains more than 24 lines. This means that the payment changes more than 24 times during the term of the loan. The loan is calculated properly and the results are correct. However, only the first 24 lines of the payment schedule are printed to the specified output file.

This type of loan can be calculated but it is necessary to request an amortization schedule to view all of the payments. For this type of loan, it is recommended that the amortization schedule be incorporated into the loan documents because of the complexity of the payment schedule.

The third type of error message is generated by the single-payment loan module. These are numbered from 150 to 179.

The fourth type of error message is generated by the revolving credit module. These are numbered from 180 to 199.

The fifth type of error message is generated by the dealer reserve routine. These are numbered from 120 through 149. In this module an error can occur because of an error in the input data or because a computed result falls outside of the tolerance specified. See the Dealer Reserve Routine for more information.

The Engine always returns the error number on line 4 of the output file, and the error message on line 5 (See Output file format below). If no error occurred, a zero is returned as the error number and the error message line is empty.

A list of all error numbers and messages can be printed by using the "-e" option on the command line.

LoanMaker SP Engine - Version 1.42

Licensed to:

Copyright 1995-99 PROM Software, Burlington VT USA - All Rights Reserved

Usage: SP_ENG [-option [-option [...]]] infile outfile

where (infile) is name of input file.

(outfile) is name of output file to hold results.

- D** dumps the input values from infile to 'stdout'.
- M** prevents alternate payment options from being computed.
- R** dumps a list of dealer reserve type numbers & descriptions.
- T** dumps a list of legal tags to 'stdout'.
- V** dumps all version numbers to 'stdout'.
- cname** specifies a different configuration file (default is "LMSP.DAT").
- dST** specifies default plan for Installment Loans, e.g., -dNY.
- e** dumps a list of all error messages to 'stdout'.
- iname** dumps insurance plan with "name", e.g.,
"-iME" dumps the "ME" insurance plan and "-iALL"
dumps all insurance plans available.
- u** remove limit on maximum number of days to first payment.
- y** show insurance cost per year (default is per day).

LoanMaker SP errors:

- 1 First due date is an invalid date. (1)**
- 2 First due date on or before Loan date. (2)**
- 3 Too many days to first payment. (3)**
- 4 Loan date is an invalid date. (4)**
- 5 Interest rate too high or too low. (5)**
- 6 There are too few payments in the loan. (6)**
- 7 There are too many payments in the loan. (7)**
- 8 The first payment cannot be skipped. (8)**
- 9 The last payment cannot be skipped. (9)**
- 10 No payments left to compute. All are specified. (10)**
- 11 Mnth specified for skip/pickup payment illegal. (11)**
- 12 Payment frequency is illegal (1, 2, 4, 12, 26, 52 OK). (12)**
- 13 Specifier 'F' cannot be a skipped payment. (13)**
- 14 Specifier '@' cannot be a skipped payment. (14)**
- 15 Unknown payment specifier. (15)**
- 16 Skip/Pick-up payment amount cannot be negative. (16)**
- 17 Prepaid fees too high in relation to the loan amount. (17)**
- 18 The computed payment is too small. (18)**
- 19 Can't find the A.P.R. (19)**
- 20 No insurance plan available for this State abbreviation. (20)**
- 21 No State specified for insurance plan. (21)**
- 22 Can't solve the loan. (22)**
- 23 Life not available on irregular loans in this state. (23)**
- 24 Life not available on balloon loans in this state. (24)**
- 25 Loan amount and/or fees cannot be negative. (25)**
- 26 Prepaid charge cannot be negative. (26)**
- 27 Note date is after loan date. (27)**
- 28 Note date is more than 60 days prior to loan date. (28)**
- 29**
- 30 Life code is illegal (0, 1 or 2). (30)**
- 31 Life term is longer than loan term (31)**
- 32 Life term is longer than mx allowed. (32)**

- 33 Truncated life not available in this state. (33)
- 34 Unknown life type. (34)
- 35 Disability code is illegal (0, 1, 2, 3 or 4). (35)
- 36 Disability term is longer than loan term (36)
- 37 Disability term is longer than max allowed. (37)
- 38 Truncated disability not available in this state. (38)
- 39 Truncated disability not available on irregular loans. (39)
- 40 Note date is an invalid date. (40)
- 41 'R' & 'M' must specify 1 or more payments, e.g. R2, R12. (41)
- 42 Too many repeat payments with 'R' or 'M' specifier. (42)
- 43 A non-zero payment amount has no effect with the 'M' specifier. (43)
- 60 IUI code is illegal (0 or 1). (60)
- 61 IUI term is longer than loan term (61)
- 62 IUI term is longer than max allowed. (62)
- 63 Truncated IUI not available. (63)
- 64 IUI not available on balloon loans. (64)
- 65 Truncated IUI not available on irregular loans. (65)
- 70 Property insurance code is illegal (0 or 1). (70)
- 71 Property insurance term is longer than loan term (71)
- 72 Property insurance term is loner than max allowed. (72)
- 73 Truncated property insurance not available. (73)
- 74 Property insurance not available on balloon loans. (74)
- 75 Truncated property insurance not available on irregular loans. (75)

100 Payment schedule has more than 24 lines. (100)

LoanMaker Dealer Reserve errors:

- 120 Too many or too few payments. (120)
- 121 Specified interest rate out of range. (121)
- 122 Specified buy rate out of range. (122)
- 123 Reserve percent paid (0.00%) too high or too low. (123)
- 124 Unknown dealer reserve type (0) specified. (124)
- 125 Payment schedule not equal to total of payments. (125)
- 126 Amount financed plus finance charge not equal to total of payments. (126)
- 127 Amount financed plus prepaid charges not equal to note amount. (127)
- 128 Interest rate more than 0.000% low for specified interest charge. (128)
- 129 Interest rate more than 0.000% high for specified interest charge. (129)
- 130 A. P. R. more than 0.000% low for specified finance charge. (130)
- 131 A. P. R. more than 0.000% high for specified finance charge. (131)
- 132 Balloon payment not allowed with reserve type 11. (132)
- 133 Balloon payment not allowed with reserve type 12. (133)
- 134 Truncated reserve term required for reserve type 0. (134)
- 135 Truncated reserve term not allowed for reserve type 0. (135)
- 136 Truncated term must be less than loan term (136)
- 137 Loan date () is an invalid date. (137)
- 138 First due date () is an invalid date. (138)
- 139 First due date is on or before the loan date. (139)
- 140 There are too many days to the first payment. (140)
- 141 Payment frequency (0) is illegal. (141)

LoanMaker Single Payment Loan errors:

- 150 Payment due date is an invalid date. (150)
- 151 Payment due date on or before Loan date. (151)
- 152 Loan date is an invalid date. (152)
- 153 Interest rate too high or too low. (153)

- 154 Prepaid fees too high in relation to loan amount. (154)
- 155 No plan available for this State abbreviation. (155)
- 156 Unknown life type for single-pay loan. (156)
- 157 Note date is after loan date. (157)
- 158 Note date is more than 60 days prior to loan date. (158)
- 159 Term too long for single-pay loan. (159)
- 160 Life code is illegal (0, 1 or 2). (160)
- 161 Note date is an invalid date. (161)
- 162 Can't find the A.P.R. (162)
- 163 Prepaid charge cannot be negative. (163)
- 164 Loan amount and/or fees cannot be negative. (164)

LoanMaker Revolving Credit errors:

- 180 Life code is Illegal (0, 1 or 2). (180)
- 181 Disability code is illegal (0, 1 or 2). (181)
- 182 IUI code is illegal (0 or 1). (182)

LoanMaker Qualify errors:

- 200 Too many or too few payments. (200)
- 201 Specified interest rate out of range. (201)
- 202 Payment frequency (0) is illegal. (202)

Engine errors:

- 209 EE 0: Can't open configuration file 'LMSP.DAT'.
- 210 EE 0: Can't open input file ''.
- 211 EE 0: Can't understand passed tag ''.
- 212 EE 0: Value passed is too long (over 40 bytes) for tag ''.
- 213 EE 0: Tag passed is too long (over 40 bytes).
- 214 EE 0: Irregular payment specifier #0 is illegal.
- 215 EE 0: Can't open file '' for amortization schedule.
- 216 EE 0: Can't open file '' for disclosure.
- 217 EE 0: Can't open file '' for life schedule.
- 218 EE 0: Unknown loan type 'x'.

Notes:

Notes: