

FEB 2 2004

SECURITIES AND EXCHANGE COMMISSION
WASHINGTON, DC 20549



04007189

FORM SE

FORM FOR SUBMISSION OF PAPER FORMAT EXHIBITS
BY ELECTRONIC FILERS

PROCESSED

FEB 03 2004

THOMSON
FINANCIAL

Structured Asset Securities Corporation
(Exact Name of Registrant as Specified in Charter)

0000808851
(Registrant CIK Number)

Form 8-K for January 27, 2004
(Electronic Report, Schedule or Registration Statement
of Which the Documents Are a Part
(Give Period of Report))

333-106925
(SEC File Number, if Available)

N/A

(Name of Person Filing the Document (if Other Than the Registrant))

SIGNATURES

Filings Made by the Registrant. The registrant has duly caused this form to be signed on its behalf by the undersigned, thereunto duly authorized, in the City of New York, State of New York, on February 2, 2004.

STRUCTURED ASSET SECURITIES CORPORATION

By: 
Name: Ellen V. Kiernan
Title: Senior Vice President

Exhibit Index

<u>Exhibit</u>	<u>Page</u>
99.1 Computational Materials.....	5

IN ACCORDANCE WITH RULE 311 (h) OF REGULATION S-T, THESE
COMPUTATIONAL MATERIALS ARE BEING FILED IN PAPER.

COMPUTATIONAL MATERIALS

for

STRUCTURED ASSET INVESTMENT LOAN TRUST

Mortgage Pass-Through Certificates, Series 2004-1

! SAIL_04_BC1_2.CDI #CMOVER_3.0D ASSET_BACKED_HOMEEQUITY PORTFOLIO !
MAX_CF_VECTSIZE 670

!
!! Created by Intex Deal Maker v3.6.092 , subroutines 3.0f_p2
!! 01/23/2004 10:39 AM

!
! Modeled in the Intex CMO Modeling Language, (NYFI4W903561)
! which is copyright (c) 2004 by Intex Solutions, Inc.
! Intex shall not be held liable for the accuracy of this data
! nor for the accuracy of information which is derived from this data.

! COLLAT_GROUPS 1 2
GROUP 1 = "1A" "1F"
GROUP 2 = "2F" "2A"

!
DEFINE PREPAY PPC GROUP "1A" RISE_PERS 12 START_CPR 27 END_CPR 27
DEFINE PREPAY PPC GROUP "1F" RISE_PERS 12 START_CPR 23 END_CPR 23
DEFINE PREPAY PPC GROUP "2F" RISE_PERS 12 START_CPR 23 END_CPR 23
DEFINE PREPAY PPC GROUP "2A" RISE_PERS 12 START_CPR 27 END_CPR 27

!
DEFINE CONSTANT #OrigCollBal = 2493049537.86
DEFINE CONSTANT #OrigCollBal1 = 1761613261.31
DEFINE CONSTANT #OrigCollBal1A = 1258652015.04
DEFINE CONSTANT #OrigCollBal1F = 502961246.27
DEFINE CONSTANT #OrigCollBal2 = 731436276.55
DEFINE CONSTANT #OrigCollBal2F = 249960249.12
DEFINE CONSTANT #OrigCollBal2A = 481476027.43

!
DEFINE CONSTANT #OrigBondBal = 2480584999.99
DEFINE CONSTANT #OrigBondBal1 = 1761613261.31
DEFINE CONSTANT #OrigBondBal2 = 731436276.55

!
DEFINE CONSTANT #SpecSenEnhPct = 22.700924898149%
DEFINE CONSTANT #SNRTargPct = 77.299075101851%
DEFINE CONSTANT #M1TargPct = 86.799627896513%
DEFINE CONSTANT #M2TargPct = 92.299952158034%
DEFINE CONSTANT #M3TargPct = 93.799967658809%
DEFINE CONSTANT #M4TargPct = 95.299983159584%
DEFINE CONSTANT #M5TargPct = 96.799998660360%
DEFINE CONSTANT #M6TargPct = 97.799955509141%
DEFINE CONSTANT #BTargPct = 99.000000000000%
DEFINE #BondBal = 2480584999.99
DEFINE #BondBal1 = 1561670000.00
DEFINE #BondBal2 = 648423000.00

!
FULL_DEALNAME: Structured Asset Investment Loan Trust 2004-BC1
SERVICER_MASTER: Aurora Loan Services

!
ISSUER: SAIL 2004-BC1
DEALER: Lehman Brothers
DEAL SIZE: \$ 2480584999.99
PRICING SPEED: GROUP "1A" PPC 100%
PRICING SPEED: GROUP "1F" PPC 100%
PRICING SPEED: GROUP "2F" PPC 100%
PRICING SPEED: GROUP "2A" PPC 100%

```

!   ISSUE DATE:    20031201
      SETTLEMENT DATE: 20031230
!
!   Record date delay: 24
!
!   DEFINE TR_INDEXDEPS_ALL
!
!   DEFINE TRANCHE "CAP_IN", "AIO1", "AIO2", "1A1", "1A2", "2A1", "2A2", "M1", "M2", "M3", "M4", "M5",
" M6", "B", "OC"
!
!   DEFINE SCHEDULE "AIO1","AIO2","AIO1_BAL","AIO2_BAL"
!
!   DEAL_CLOCK_INFO_
      ISSUE_CDU_DATE    20031201_
      DEAL_FIRSTPAY_DATE    20040125
!
!
!   DEFINE TABLE "CapNotional" (25, 2) = "CURDATE" "Balance"
      20040125.1    0
      20040225.1    1889183100
      20040325.1    1873704161
      20040425.1    1853010612
      20040525.1    1829521766
      20040625.1    1802880790
      20040725.1    1772995397
      20040825.1    1738810083
      20040925.1    1700358666
      20041025.1    1656315082
      20041125.1    1588363683
      20041225.1    1523408466
      20050125.1    1465135820
      20050225.1    1412238512
      20050325.1    1363635315
      20050425.1    1317081659
      20050525.1    1272409763
      20050625.1    1229616384
      20050725.1    1190718586
      20050825.1    1155138390
      20050925.1    1121849014
      20051025.1    1083602470
      20051125.1    945143070.1
      20051225.1    821971679.9
      20060125.1    0
!
!   DEFINE DYNAMIC #CapBal = LOOKUP_TBL( "STEP", Curdate, "CapNotional", "CURDATE", "Balance"
)
!
!   DEFINE DYNAMIC #CapBalEnd = LOOKUP_TBL( "STEP", Curdate + 30, "CapNotional", "CURDATE",
"Balance" )
!
!
!   DEFINE #FloorFixed    = 12464537.86
!   DEFINE #ReqPerc        = 0
!   DEFINE #TrigEnhFrac    = 0
!   DEFINE #CumLossShft    = 0

```

```

DEFINE #TrigCumLossFrac = 0
DEFINE #SDCumLossShft = 0
DEFINE #SDTrigCumLossFrac = 0
DEFINE #SpecOCTarg = 12464537.86
DEFINE STANDARDIZE OC_ACTUAL_VAL #OC = 12464537.87
DEFINE STANDARDIZE OCT_INITVAL CONSTANT #InitOCTarg = 12464537.86
DEFINE STANDARDIZE OCT_STEPDOWN_MONTH CONSTANT #StepDownDate = 37
DEFINE STANDARDIZE OCT_STEPDOWN_FRAC CONSTANT #StepOCFrac = 0.01
DEFINE STANDARDIZE EXCESS_INTEREST #XSSpread = 0
DEFINE STANDARDIZE OCT_FLOOR CONSTANT #FloorOCTarg = #FloorFixed
DEFINE STANDARDIZE OCT_VAL DYNAMIC #Octval = #SpecOCTarg
!
DEFINE DYNAMIC STICKY #NetRate = ( COLL_I_MISC("COUPON") ) / COLL_PREV_BAL * 1200
DEFINE DYNAMIC STICKY #NetRate1 = ( COLL_I_MISC("COUPON", 1) ) / COLL_PREV_BAL(1) *
1200
DEFINE DYNAMIC STICKY #NetRate2 = ( COLL_I_MISC("COUPON", 2) ) / COLL_PREV_BAL(2) *
1200
!
DEFINE DYNAMIC STICKY #NetRateActual360 = #Netrate * 30 / DAYS_DIFF(CURDATE ,
MONTHS_ADD(CURDATE,-1))
!
!
DEFINE TABLE "OC_SDCUMLOSS0" (38, 2) = "MONTH" "OC_SDCUMLOSS_FRAC0"
37.1 0.03
38.1 0.030625
39.1 0.03125
40.1 0.031875
41.1 0.0325
42.1 0.033125
43.1 0.03375
44.1 0.034375
45.1 0.035
46.1 0.035625
47.1 0.03625
48.1 0.036875
49.1 0.0375
50.1 0.038125
51.1 0.03875
52.1 0.039375
53.1 0.04
54.1 0.040625
55.1 0.04125
56.1 0.041875
57.1 0.0425
58.1 0.043125
59.1 0.04375
60.1 0.044375
61.1 0.045
62.1 0.04520833333
63.1 0.04541666667
64.1 0.045625
65.1 0.04583333333
66.1 0.04604166667
67.1 0.04625
68.1 0.04645833333

```

69.1 0.04666666667
70.1 0.046875
71.1 0.04708333333
72.1 0.04729166667
73.1 0.0475
360.1 0.0475

! DEFINE TABLE "OC_CUMLOSS0" (38, 2) = "MONTH" "OC_CUMLOSS_FRAC0"

37.1 0.03
38.1 0.030625
39.1 0.03125
40.1 0.031875
41.1 0.0325
42.1 0.033125
43.1 0.03375
44.1 0.034375
45.1 0.035
46.1 0.035625
47.1 0.03625
48.1 0.036875
49.1 0.0375
50.1 0.038125
51.1 0.03875
52.1 0.039375
53.1 0.04
54.1 0.040625
55.1 0.04125
56.1 0.041875
57.1 0.0425
58.1 0.043125
59.1 0.04375
60.1 0.044375
61.1 0.045
62.1 0.04520833333
63.1 0.04541666667
64.1 0.045625
65.1 0.04583333333
66.1 0.04604166667
67.1 0.04625
68.1 0.04645833333
69.1 0.04666666667
70.1 0.046875
71.1 0.04708333333
72.1 0.04729166667
73.1 0.0475
360.1 0.0475

! DEFINE DYNAMIC #AIO1_SCHED = SCHED_AMOUNT("AIO1")
! DEFINE DYNAMIC #AIO1_ENDBAL = SCHED_AMOUNT("AIO1_BAL")

! DEFINE DYNAMIC #AIO2_SCHED = SCHED_AMOUNT("AIO2")
! DEFINE DYNAMIC #AIO2_ENDBAL = SCHED_AMOUNT("AIO2_BAL")

! TOLERANCE WRITEDOWN_0LOSS 1.00

```

!
DEFINE DYNAMIC STICKY #SubBalGr1 = MAX(0, COLL_PREV_BAL(1) - BBAL("1A1","1A2"))
DEFINE DYNAMIC STICKY #SubBalGr2 = MAX(0, COLL_PREV_BAL(2) - BBAL("2A1","2A2"))
!
DEFINE DYNAMIC STICKY #NetFundsCap1 = (COLL_I_MISC("COUPON",1) -
OPTIMAL_INTPMT("AIO1#1")) / COLL_PREV_BAL(1) * 1200
DEFINE DYNAMIC STICKY #NetFundsCap2 = (COLL_I_MISC("COUPON",2) -
OPTIMAL_INTPMT("AIO2#1")) / COLL_PREV_BAL(2) * 1200
!
DEFINE DYNAMIC STICKY #NetFundsCapSub = (#NetFundsCap1 * #SubBalGr1 + #NetFundsCap2 *
#SubBalGr2) / (#SubBalGr1 + #SubBalGr2)
!
DEFINE DYNAMIC STICKY #Cap1 = IF BBAL("2A1","2A2") > 0.01 THEN #NetFundsCap1 ELSE
#NetFundsCapSub
DEFINE DYNAMIC STICKY #Cap2 = IF BBAL("1A1","1A2") > 0.01 THEN #NetFundsCap2 ELSE
#NetFundsCapSub
!
INITIAL INDEX LIBOR_6MO 1.12
INITIAL INDEX LIBOR_1MO 1.12
!
!
Tranche "CAP_IN" PSEUDO HEDGE
Block $ 0.01 at 0.02 FLOAT NOTIONAL WITH FORMULA BEGIN ( #CapBal ); END ( #CapBalEnd ); _
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
Delay 0 Dated 20031225 Next 20040125
((1 * MIN(5.10, LIBOR_1MO)) + (-1.10))
0 999
!
Tranche "AIO1" SEN_IO
Block 422787182.71 at 6 GROUP 1 FREQ M NOTIONAL WITH FORMULA BEGIN ( IF CURMONTH
LE 18 THEN (#AIO1_SCHED) ELSE 0 ); _
END ( IF CURMONTH LT 18 THEN (#AIO1_ENDBAL) ELSE 0 ); _
DAYCOUNT 30360 BUSINESS_DAY NONE _
Delay 0 Dated 20031225 Next 20040125
!
Tranche "AIO2" SEN_IO
Block 175544706.37 at 6 GROUP 2 FREQ M NOTIONAL WITH FORMULA BEGIN ( IF CURMONTH
LE 18 THEN (#AIO2_SCHED) ELSE 0 ); _
END ( IF CURMONTH LT 18 THEN (#AIO2_ENDBAL) ELSE 0 ); _
DAYCOUNT 30360 BUSINESS_DAY NONE _
Delay 0 Dated 20031225 Next 20040125
!
Tranche "1A1" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 1163446000.00 at 1.33 GROUP 1 FREQ M FLOAT RESET M _
COUPONCAP 30360 NONE ( #Cap1 ); _
DAYCOUNT 30360 BUSINESS_DAY NONE _
Delay 0 Dated 20031225 Next 20040125
(1 * LIBOR_1MO + ( IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 0.42 ELSE
0.21 ))
0 999
!
Tranche "1A2" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 398224000.00 at 2.406 GROUP 1 FREQ M FLOAT RESET M _
COUPONCAP 30360 NONE ( #Cap1 ); _
DAYCOUNT 30360 BUSINESS_DAY NONE _

```

Delay 0 Dated 20031225 Next 20040125
 (1 * LIBOR_1MO + (IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 0.8 ELSE
 0.55))
 20040125 2.406 2.406
 20060125 0 999
 99999999
 !
 Tranche "2A1" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 480073000.00 at 1.32 GROUP 2 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE (#Cap2); _
 DAYCOUNT 30360 BUSINESS_DAY NONE _
 Delay 0 Dated 20031225 Next 20040125
 (1 * LIBOR_1MO + (IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 0.4 ELSE
 0.2))
 0 999
 !
 Tranche "2A2" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 168350000.00 at 2.406 GROUP 2 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE (#Cap2); _
 DAYCOUNT 30360 BUSINESS_DAY NONE _
 Delay 0 Dated 20031225 Next 20040125
 (1 * LIBOR_1MO + (IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 0.8 ELSE
 0.55))
 20040125 2.406 2.406
 20060125 0 999
 99999999
 !
 Tranche "M1" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 118420000.00 at 1.77 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE (#NetFundsCapSub); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20031225 Next 20040125
 (1 * LIBOR_1MO + (IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 0.975 ELSE
 0.65))
 0 999
 !
 Tranche "M2" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 68559000.00 at 2.92 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE (#NetFundsCapSub); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20031225 Next 20040125
 (1 * LIBOR_1MO + (IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 2.7 ELSE
 1.8))
 0 999
 !
 Tranche "M3" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 18697000.00 at 3.07 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE (#NetFundsCapSub); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20031225 Next 20040125
 (1 * LIBOR_1MO + (IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 2.925 ELSE
 1.95))
 0 999
 !
 Tranche "M4" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);

```

Block 18697000.00 at 3.87 FREQ M FLOAT RESET M _
  COUPONCAP 30360 NONE ( #NetFundsCapSub ); _
  DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
  Delay 0 Dated 20031225 Next 20040125
  (1 * LIBOR_1MO + ( IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 4.125 ELSE
2.75 ))
0 999
!
Tranche "M5" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 18697000.00 at 4.12 FREQ M FLOAT RESET M _
  COUPONCAP 30360 NONE ( #NetFundsCapSub ); _
  DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
  Delay 0 Dated 20031225 Next 20040125
  (1 * LIBOR_1MO + ( IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 4.5 ELSE 3
))
0 999
!
Tranche "M6" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 12464000.00 at 4.12 FREQ M FLOAT RESET M _
  COUPONCAP 30360 NONE ( #NetFundsCapSub ); _
  DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
  Delay 0 Dated 20031225 Next 20040125
  (1 * LIBOR_1MO + ( IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 4.5 ELSE 3
))
0 999
!
Tranche "B" JUN_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 14957999.99 at 6 FREQ M FLOAT _
  COUPONCAP 30360 NONE ( #NetFundsCapSub ); _
  DAYCOUNT 30360 BUSINESS_DAY NONE _
  Delay 24 Dated 20031201 Next 20040125
  ( IF ((COLL_PREV_BAL("LAGMON_1") / #OrigCollBal) < 10%) THEN 6.5 ELSE 6 )
0 999
!
Tranche "R" SEN_RES
Block 2493049537.86 at 0 NOTIONAL WITH GROUP 0 SURPLUS _
  DAYCOUNT 30360 BUSINESS_DAY NONE _
  FREQ M Delay 24 Dated 20031201 Next 20040125
!
Tranche "OC" SEN_OC_RES
Block 12464537.87 at 0 _
  DAYCOUNT 30360 BUSINESS_DAY NONE _
  FREQ M Delay 24 Dated 20031201 Next 20040125
!
Tranche "#OC" SYMVAR
Tranche "#SpecOCTarg" SYMVAR
!
!
DEFINE PSEUDO_TRANCHE COLLAT _
  Delay 24 Dated 20031201 Next 20040125 Settle 20031230
DEFINE PSEUDO_TRANCHE COLLAT GROUP 1 _
  Delay 24 Dated 20031201 Next 20040125 Settle 20031230
DEFINE PSEUDO_TRANCHE COLLAT GROUP 2 _
  Delay 24 Dated 20031201 Next 20040125 Settle 20031230
!

```

HEDGE "Cap" _
TYPE CAP _
LEG "FLT" DEAL_RECEIVES OPTIMAL_INTPMT "CAP_IN" _
BY "Lehman Brothers Special Financing"

!

CLASS "X" = "AIO1" "AIO2"
CLASS "1A1" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "1A1"
CLASS "1A2" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "1A2"
CLASS "2A1" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "2A1"
CLASS "2A2" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "2A2"
CLASS "M1" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "M1"
CLASS "M2" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "M2"
CLASS "M3" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "M3"
CLASS "M4" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "M4"
CLASS "M5" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "M5"
CLASS "M6" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "M6"
CLASS "B" NO_BUILD_TRANCHE _
SHORTFALL_PAYBACK COUPONCAP TRUE _
SHORTFALL_EARN_INT COUPONCAP TRUE _
= "B"
CLASS "RESID" = "R#1" "OC#1"
CLASS "SNR_1" ALLOCATION _
= "1A1" "1A2"
CLASS "SNR_2" ALLOCATION _

```

    = "2A1" "2A2"
CLASS "SNR" DISTRIB_CLASS PRORATA WRITEDOWN_BAL PRORATA ALLOCATION _
    = "SNR_1" "SNR_2"
!
!
CLASS "ROOT"
    WRITEDOWN_BAL RULES _
    DISTRIB_CLASS RULES _
    SHORTFALL_PAYBACK PRINCIPAL_LOSS TRUE _
    SHORTFALL_EARN_INT INTEREST TRUE _
    = "X" "SNR" "M1" "M2" "M3" "M4" "M5" "M6" "B" "RESID"
!
    DEFINE PSEUDO_TRANCHE CLASS "SNR"          Delay 24 Dated 20031201 Next 20040125
DAYCOUNT 30360 BUSINESS_DAY NONE
!
    DEFINE PSEUDO_TRANCHE CLASS "SNR_1"        Delay 24 Dated 20031201 Next 20040125
DAYCOUNT 30360 BUSINESS_DAY NONE
!
    DEFINE PSEUDO_TRANCHE CLASS "SNR_2"        Delay 24 Dated 20031201 Next 20040125
DAYCOUNT 30360 BUSINESS_DAY NONE
!
!
CROSSOVER When 0
!
!
TRIGGER "StepUp-CumLoss" _
    FULL_NAME "Step Up Cumulative Loss Trigger" _
    ORIG_TESTVAL 0.000% _
    TESTVAL (#TrigCumLossFrac); _
    ORIG_TARGETVAL 3% _
    TARGETVAL (#CumLossShft); _
    TRIGVAL LODIFF
!
!
TRIGGER "StepUp-DlqEnh" _
    FULL_NAME "Step Up Enhancement Delinquency Trigger" _
    ORIG_TESTVAL 0.000% _
    TESTVAL (#TrigEnhFrac); _
    ORIG_TARGETVAL 7.49087862682042% _
    TARGETVAL (#ReqPerc); _
    TRIGVAL LODIFF
!
!
TRIGGER "STEPUP_TRIGGER" _
    FULL_NAME "Step Up Trigger" _
    DEFINITION "A Step Up Trigger exists, if
;(1) a percentage calculated as the quotient of the amount of cumulative
realized losses divided by the original collateral balance exceeds the target defined by a schedule;
    Month <= %; _
    37 3%; _
    38 3.0625%; _
    39 3.125%; _
    40 3.1875%; _
    41 3.25%; _
    42 3.3125%; _
    43 3.375%; _
    44 3.4375%; _
    45 3.5%; _

```

46	3.5625%; _
47	3.625%; _
48	3.6875%; _
49	3.75%; _
50	3.8125%; _
51	3.875%; _
52	3.9375%; _
53	4%; _
54	4.0625%; _
55	4.125%; _
56	4.1875%; _
57	4.25%; _
58	4.3125%; _
59	4.375%; _
60	4.4375%; _
61	4.5%; _
62	4.520833333%; _
63	4.541666667%; _
64	4.5625%; _
65	4.583333333%; _
66	4.604166667%; _
67	4.625%; _
68	4.645833333%; _
69	4.666666667%; _
70	4.6875%; _
71	4.708333333%; _
72	4.729166667%; _
73	4.75%; _
360	4.75%; _

or;(2) the aggregate principal balance of all delinquent loans * 1_
as a percentage of the respective collateral balance exceeds :_
0.66 * the Senior Enhancement Percentage."

IMPACT "If a Step Up Trigger is in effect the OC target will change to_
the last value before the trigger occurred if a stepdown has_
occurred. It has no effect if a stepdown has not occurred."

TRIGVAL FORMULA (min(TRIGGER("StepUp-CumLoss","TRIGVAL"),
TRIGGER("StepUp-DlqEnh","TRIGVAL")));

!

TRIGGER "StepDown-CumLoss" _

FULL_NAME "Step Down Cumulative Loss Trigger" _

ORIG_TESTVAL 0.000% _

TESTVAL (#SDTrigCumLossFrac); _

ORIG_TARGETVAL 3% _

TARGETVAL (#SDCumLossShft); _

TRIGVAL LODIFF

!

TRIGGER "STEPDOWN_TRIGGER" _

FULL_NAME "Step Down Trigger" _

DEFINITION "A Step Down Trigger exists, if _

;(1) a percentage calculated as the quotient of the amount of cumulative_
realized losses divided by the collateral balance exceeds the target defined by a schedule; _

Month <= %; _

37 3%; _

38 3.0625%; _

39	3.125%; _
40	3.1875%; _
41	3.25%; _
42	3.3125%; _
43	3.375%; _
44	3.4375%; _
45	3.5%; _
46	3.5625%; _
47	3.625%; _
48	3.6875%; _
49	3.75%; _
50	3.8125%; _
51	3.875%; _
52	3.9375%; _
53	4%; _
54	4.0625%; _
55	4.125%; _
56	4.1875%; _
57	4.25%; _
58	4.3125%; _
59	4.375%; _
60	4.4375%; _
61	4.5%; _
62	4.520833333%; _
63	4.541666667%; _
64	4.5625%; _
65	4.583333333%; _
66	4.604166667%; _
67	4.625%; _
68	4.645833333%; _
69	4.666666667%; _
70	4.6875%; _
71	4.708333333%; _
72	4.729166667%; _
73	4.75%; _
360	4.75%; _

"
 - IMPACT "If a Step Down Trigger is in effect the OC target CANNOT stepdown to_
 1% of the total current balance of all bonds." _

TRIGVAL FORMULA (TRIGGER("StepDown-CumLoss","TRIGVAL"));

!

OPTIONAL REDEMPTION: "Cleanup" _

WHEN_EXPR (COLL_PREV_BAL / #OrigCollBal < 10%); _

PRICE_P (COLL_BAL); _

PRICE_I (COLL_BAL * COLL_NETRATE / 1200 * 24 / 30); _

DISTR_I ACCRUE

!

OPTIONAL REDEMPTION "DealTermination" _

MANDATORY _

PRICE_P (COLL_BAL); _

WHEN_EXPR (DBAL LT .01);

!

!

INTEREST_SHORTFALL GROUP 1 FULL_PREPAY Compensate Pro_rata _

PARTIAL_PREPAY Compensate Pro_rata _

```

                LOSS      Compensate Pro_rata
!
INTEREST_SHORTFALL GROUP 2 FULL_PREPAY  Compensate Pro_rata _
                PARTIAL_PREPAY Compensate Pro_rata _
                LOSS      Compensate Pro_rata
!
DEFINE MACRO BLOCK #SNR_Int =
{
-----
    from : CLASS ( "SNR" )
    pay : CLASS INTEREST PRO_RATA ( "SNR_1"; "SNR_2" )
-----
    from : CLASS ( "SNR_1" )
    pay : CLASS INTEREST PRO_RATA ( "1A1"; "1A2" )
-----
    from : CLASS ( "SNR_2" )
    pay : CLASS INTEREST PRO_RATA ( "2A1"; "2A2" )
-----
}
DEFINE MACRO BLOCK #SNR_InS =
{
-----
    from : CLASS ( "SNR" )
    pay : CLASS INTSHORT PRO_RATA ( "SNR_1"; "SNR_2" )
-----
    from : CLASS ( "SNR_1" )
    pay : CLASS INTSHORT PRO_RATA ( "1A1"; "1A2" )
-----
    from : CLASS ( "SNR_2" )
    pay : CLASS INTSHORT PRO_RATA ( "2A1"; "2A2" )
-----
}
DEFINE MACRO BLOCK #SNR_Prn[1] =
{
-----
    when : IS_TRUE( {#1} )
    calculate : #SeniorPrinc      = #PrincPmt / #DistribAmt * #ClassSNRPDA
    calculate : #SeniorXtraP      = #ClassSNRPDA - #SeniorPrinc
!
    calculate : #SeniorPDA1      = MIN( BBAL("SNR_1"), (#SeniorPrinc * #PrincFrac1) + (#SeniorXtraP *
#XtraPFrac1))
    calculate : #SeniorPDA2      = MIN( BBAL("SNR_2"), (#SeniorPrinc * #PrincFrac2) + (#SeniorXtraP *
#XtraPFrac2))
-----
    from : SUBACCOUNT ( #SeniorPDA1, CLASS "SNR" )
    pay : CLASS BALANCE SEQUENTIAL ( "SNR_1" )
-----
    from : SUBACCOUNT ( #SeniorPDA2, CLASS "SNR" )
    pay : CLASS BALANCE SEQUENTIAL ( "SNR_2" )
-----
    from : CLASS ( "SNR" )

```

```

    pay : CLASS BALANCE PRO_RATA ( "SNR_1"; "SNR_2" )
-----
!
    calculate : #P_SNR_1 = BBAL("1A1","1A2") - BBAL("SNR_1")
-----
        from : CLASS ( "SNR_1" )
        pay : CLASS INTEREST SEQUENTIAL ( "1A1" )
        pay : CLASS INTSHORT SEQUENTIAL ( "1A1" )
-----
        from : CLASS ( "SNR_1" )
        from : SUBACCOUNT ( #P_SNR_1 )
        pay : CLASS BALANCE SEQUENTIAL ( "1A1" )
-----
        from : CLASS ( "SNR_1" )
        pay : CLASS INTEREST SEQUENTIAL ( "1A2" )
        pay : CLASS INTSHORT SEQUENTIAL ( "1A2" )
-----
        from : CLASS ( "SNR_1" )
        from : SUBACCOUNT ( #P_SNR_1 )
        pay : CLASS BALANCE SEQUENTIAL ( "1A2" )
-----
!
    calculate : #P_SNR_2 = BBAL("2A1","2A2") - BBAL("SNR_2")
-----
        from : CLASS ( "SNR_2" )
        pay : CLASS INTEREST SEQUENTIAL ( "2A1" )
        pay : CLASS INTSHORT SEQUENTIAL ( "2A1" )
-----
        from : CLASS ( "SNR_2" )
        from : SUBACCOUNT ( #P_SNR_2 )
        pay : CLASS BALANCE SEQUENTIAL ( "2A1" )
-----
        from : CLASS ( "SNR_2" )
        pay : CLASS INTEREST SEQUENTIAL ( "2A2" )
        pay : CLASS INTSHORT SEQUENTIAL ( "2A2" )
-----
        from : CLASS ( "SNR_2" )
        from : SUBACCOUNT ( #P_SNR_2 )
        pay : CLASS BALANCE SEQUENTIAL ( "2A2" )
-----
!
        from : CLASS ( "1A1" )
        pay : SEQUENTIAL ( "1A1#1" )
-----
        from : CLASS ( "1A2" )
        pay : SEQUENTIAL ( "1A2#1" )
-----
        from : CLASS ( "2A1" )
        pay : SEQUENTIAL ( "2A1#1" )
-----
        from : CLASS ( "2A2" )
        pay : SEQUENTIAL ( "2A2#1" )
-----
}

```

```

DEFINE MACRO BLOCK #M1_Prn =
{
-----
    from : CLASS ( "M1" )
    pay : SEQUENTIAL ( "M1#1" )
-----
}
DEFINE MACRO BLOCK #M2_Prn =
{
-----
    from : CLASS ( "M2" )
    pay : SEQUENTIAL ( "M2#1" )
-----
}
DEFINE MACRO BLOCK #M3_Prn =
{
-----
    from : CLASS ( "M3" )
    pay : SEQUENTIAL ( "M3#1" )
-----
}
DEFINE MACRO BLOCK #M4_Prn =
{
-----
    from : CLASS ( "M4" )
    pay : SEQUENTIAL ( "M4#1" )
-----
}
DEFINE MACRO BLOCK #M5_Prn =
{
-----
    from : CLASS ( "M5" )
    pay : SEQUENTIAL ( "M5#1" )
-----
}
DEFINE MACRO BLOCK #M6_Prn =
{
-----
    from : CLASS ( "M6" )
    pay : SEQUENTIAL ( "M6#1" )
-----
}
DEFINE MACRO BLOCK #B_Prn =
{
-----
    from : CLASS ( "B" )
    pay : SEQUENTIAL ( "B#1" )
-----
}
!
!
CMO Block Payment Rules
-----
calculate : #HedgePaySave = 0.00
-----

```

```

calculate : #PrincFrac1      = COLL_P(1) / COLL_P
calculate : #PrincFrac2      = COLL_P(2) / COLL_P
!
calculate : #XtraPFrac1      = COLL_P(1) / COLL_P
calculate : #XtraPFrac2      = COLL_P(2) / COLL_P
!
calculate : #Princ           = COLL_P
!
calculate : #Interest        = COLL_I
!
calculate : #PrevSpecOC      = #SpecOCTarg
!
calculate : #CurrentOC       = MAX( 0, COLL_BAL - (BBAL("1A1#1", "1A2#1", "2A1#1", "2A2#1",
"M1#1", "M2#1", "M3#1", "M4#1", "M5#1", "M6#1", "B#1") - #Princ))
!
calculate : #XSSpread        = MAX( 0, #Interest - OPTIMAL_INTPMT("ROOT") -
INTSHORT_ACCUM("SNR") + COUPONCAP_SHORTFALL("ROOT") )
!
calculate : #FloorOCTotal    = #FloorOCTarg
!
calculate : #StepOCTarg      = DBAL * #StepOCFrac
!
calculate : #StepDownDatePass = CURMONTH GE #StepDownDate
!
!!!***** BEGINNING OF SENIOR ENHANCEMENT PCT CALCULATION *****
!!! ASSUME STEPDOWN IN ORDER TO CALCULATE SENIOR ENHANCEMENT PCT
calculate : #SpecOCTarg      = MAX( MIN( #InitOCTarg, #StepOCTarg ) , #FloorOCTotal )
!
calculate : #SpecOCTarg      = MIN( #SpecOCTarg, COLL_BAL )
!
calculate : #SpecOCTarg      = #Octval
!
calculate : #OCDeficiency    = MAX(0, #SpecOCTarg - #CurrentOC)
!
calculate : #OCSurplus       = MINMAX(0, #CurrentOC - #SpecOCTarg, COLL_P)
!
calculate : #PrincPmt        = MAX(0, COLL_P - #OCSurplus)
!
calculate : #XSIntRem        = MAX( 0, #Interest - OPTIMAL_INTPMT("ROOT") -
INTSHORT_ACCUM("SNR") + #OCSurplus + COUPONCAP_SHORTFALL("ROOT"))
!
calculate : #SubDefic        = MAX ( 0, ( BBAL("ROOT") - BBAL( "OC#1" ) - #Princ ) - COLL_BAL )
!
calculate : #AddPrinc        = MIN( #XSIntRem, #SubDefic )
calculate : #XSIntRem        = MAX( 0, #XSIntRem - #AddPrinc )
!
calculate : #XtraPDA         = MIN( #OCDeficiency, #XSIntRem )
calculate : #XSIntRem        = MAX( 0, #XSIntRem - #XtraPDA )
!
calculate : #DistribAmt      = #PrincPmt + #AddPrinc + #XtraPDA
!
calculate : #ClassSNRPDA     = BBAL("1A1", "1A2", "2A1", "2A2") _
                             - MIN(COLL_BAL - #FloorOCTotal, #SNRTargPct * COLL_BAL)
calculate : #ClassSNRPDA     = MAX( 0.0, MIN(BBAL("1A1", "1A2", "2A1", "2A2"), #ClassSNRPDA
))

```

```

calculate : #ClassSNRPDA      = MAX( 0, MIN( #ClassSNRPDA, #DistribAmt ) )
!
!
!!!***** END OF SENIOR ENHANCEMENT PCT CALCULATION *****
!
calculate : #SenEnhancePct    = (COLL_BAL - (BBAL("SNR") - #ClassSNRPDA )) / COLL_BAL
!
calculate : #StepDownBal      = (#SenEnhancePct - #SpecSenEnhPct) + 1E-8 GE 0.00
!
calculate : #SDCumLossShft    = LOOKUP_TBL( "STEP", CURMONTH , "OC_SDCUMLOSS0",
"MONTH", "OC_SDCUMLOSS_FRAC0" )
calculate : #SDTrigCumLossFrac = DELINQ_LOSS_ACCUM / #OrigCollBal
!
calculate : #SDTrigEvent      = TRIGGER("STEPDOWN_TRIGGER")
!
calculate : #StepDown         = #StepDown OR (( #StepDownDatePass AND #StepDownBal ) AND
NOT #SDTrigEvent )
!
calculate : #ReqPerc          = 0.66 *(COLL_BAL - (BBAL("SNR") - #ClassSNRPDA )) / COLL_BAL
!
calculate : #TrigEnhFrac      = 1 * AVG_COLL("RATE",-1,2,1)
!
calculate : #CumLossShft      = LOOKUP_TBL( "STEP", CURMONTH , "OC_CUMLOSS0",
"MONTH", "OC_CUMLOSS_FRAC0" )
calculate : #TrigCumLossFrac  = DELINQ_LOSS_ACCUM / #OrigCollBal
!
calculate : #TrigEvent        = TRIGGER("STEPUP_TRIGGER")
!
calculate : #TrigOCTargPost    = #PrevSpecOC
!
calculate : #SpecOCTarg        = IF #StepDown _
                                THEN IF #TrigEvent _
                                    THEN MAX( MIN( #InitOCTarg, #StepOCTarg ) , #TrigOCTargPost,
#FloorOCTotal ) _
                                ELSE MAX( MIN( #InitOCTarg, #StepOCTarg ) , #FloorOCTotal ) _
                                ELSE MAX ( #InitOCTarg, #FloorOCTotal )
!
calculate : #SpecOCTarg        = MIN( #SpecOCTarg, COLL_BAL )
!
calculate : #SpecOCTarg        = #Octval
!
calculate : #OCDeficiency      = MAX(0, #SpecOCTarg - #CurrentOC)
!
calculate : #OCSurplus         = MINMAX(0, #CurrentOC - #SpecOCTarg, COLL_P)
!
calculate : #PrincPmt          = MAX(0, COLL_P - #OCSurplus)
!
!
calculate : #XSIntRem          = MAX( 0, #Interest - OPTIMAL_INTPMT("ROOT") -
INTSHORT_ACCUM("SNR") + #OCSurplus + COUPONCAP_SHORTFALL("ROOT"))
!
calculate : #SubDefic          = MAX ( 0, ( BBAL("ROOT") - BBAL( "OC#1" ) - #Princ ) - COLL_BAL )
!
calculate : #AddPrinc          = MIN( #XSIntRem, #SubDefic )
calculate : #XSIntRem          = MAX( 0, #XSIntRem - #AddPrinc )

```

```

!
calculate : #XtraPDA      = MIN( #OCDeficiency, #XSIntRem )
calculate : #XSIntRem    = MAX( 0, #XSIntRem - #XtraPDA )
!
calculate : #DistribAmt   = #PrincPmt + #AddPrinc + #XtraPDA
!
calculate : #ClassSNRPDA  = IF ( #TrigEvent OR ( #StepDown EQ 0.0 ) ) _
                          THEN #DistribAmt _
                          ELSE BBAL("1A1", "1A2", "2A1", "2A2") _
                          - MIN(COLL_BAL - #FloorOCTotal, #SNRTargPct * COLL_BAL)
calculate : #ClassSNRPDA  = MAX( 0.0, MIN(BBAL("1A1", "1A2", "2A1", "2A2"), #ClassSNRPDA
))
calculate : #ClassSNRPDA  = MAX( 0, MIN( #ClassSNRPDA, #DistribAmt ) )
!
!
calculate : #ClassM1PDA   = IF ( #TrigEvent OR ( #StepDown EQ 0.0 ) ) _
                          THEN #DistribAmt - #ClassSNRPDA _
                          ELSE BBAL("1A1", "1A2", "2A1", "2A2", "M1") - #ClassSNRPDA _
                          - MIN(COLL_BAL - #FloorOCTotal, #M1TargPct * COLL_BAL)
calculate : #ClassM1PDA   = MAX( 0.0, MIN(BBAL("M1"), #ClassM1PDA ) )
calculate : #ClassM1PDA   = MAX( 0, MIN( #ClassM1PDA, #DistribAmt - #ClassSNRPDA ) )
!
!
calculate : #ClassM2PDA   = IF ( #TrigEvent OR ( #StepDown EQ 0.0 ) ) _
                          THEN #DistribAmt - #ClassSNRPDA - #ClassM1PDA _
                          ELSE BBAL("1A1", "1A2", "2A1", "2A2", "M1", "M2") - #ClassSNRPDA -
#ClassM1PDA _
                          - MIN(COLL_BAL - #FloorOCTotal, #M2TargPct * COLL_BAL)
calculate : #ClassM2PDA   = MAX( 0.0, MIN(BBAL("M2"), #ClassM2PDA ) )
calculate : #ClassM2PDA   = MAX( 0, MIN( #ClassM2PDA, #DistribAmt - #ClassSNRPDA -
#ClassM1PDA ) )
!
!
calculate : #ClassM3PDA   = IF ( #TrigEvent OR ( #StepDown EQ 0.0 ) ) _
                          THEN #DistribAmt - #ClassSNRPDA - #ClassM1PDA - #ClassM2PDA _
                          ELSE BBAL("1A1", "1A2", "2A1", "2A2", "M1", "M2", "M3") - #ClassSNRPDA -
#ClassM1PDA - #ClassM2PDA _
                          - MIN(COLL_BAL - #FloorOCTotal, #M3TargPct * COLL_BAL)
calculate : #ClassM3PDA   = MAX( 0.0, MIN(BBAL("M3"), #ClassM3PDA ) )
calculate : #ClassM3PDA   = MAX( 0, MIN( #ClassM3PDA, #DistribAmt - #ClassSNRPDA -
#ClassM1PDA - #ClassM2PDA ) )
!
!
calculate : #ClassM4PDA   = IF ( #TrigEvent OR ( #StepDown EQ 0.0 ) ) _
                          THEN #DistribAmt - #ClassSNRPDA - #ClassM1PDA - #ClassM2PDA -
#ClassM3PDA _
                          ELSE BBAL("1A1", "1A2", "2A1", "2A2", "M1", "M2", "M3", "M4") -
#ClassSNRPDA - #ClassM1PDA - #ClassM2PDA - #ClassM3PDA _
                          - MIN(COLL_BAL - #FloorOCTotal, #M4TargPct * COLL_BAL)
calculate : #ClassM4PDA   = MAX( 0.0, MIN(BBAL("M4"), #ClassM4PDA ) )
calculate : #ClassM4PDA   = MAX( 0, MIN( #ClassM4PDA, #DistribAmt - #ClassSNRPDA -
#ClassM1PDA - #ClassM2PDA - #ClassM3PDA ) )
!
!
calculate : #ClassM5PDA   = IF ( #TrigEvent OR ( #StepDown EQ 0.0 ) ) _

```

```

THEN #DistribAmt - #ClassSNRPDA - #ClassM1PDA - #ClassM2PDA -
#ClassM3PDA - #ClassM4PDA _
ELSE BBAL("1A1", "1A2", "2A1", "2A2", "M1", "M2", "M3", "M4", "M5") -
#ClassSNRPDA - #ClassM1PDA - #ClassM2PDA - #ClassM3PDA - #ClassM4PDA _
- MIN(COLL_BAL - #FloorOCTotal, #M5TargPct * COLL_BAL)
calculate : #ClassM5PDA = MAX( 0.0, MIN(BBAL("M5"), #ClassM5PDA ))
calculate : #ClassM5PDA = MAX( 0, MIN( #ClassM5PDA, #DistribAmt - #ClassSNRPDA -
#ClassM1PDA - #ClassM2PDA - #ClassM3PDA - #ClassM4PDA ) )
!
!
calculate : #ClassM6PDA = IF (#TrigEvent OR (#StepDown EQ 0.0)) _
THEN #DistribAmt - #ClassSNRPDA - #ClassM1PDA - #ClassM2PDA -
#ClassM3PDA - #ClassM4PDA - #ClassM5PDA _
ELSE BBAL("1A1", "1A2", "2A1", "2A2", "M1", "M2", "M3", "M4", "M5", "M6") -
#ClassSNRPDA - #ClassM1PDA - #ClassM2PDA - #ClassM3PDA - #ClassM4PDA - #ClassM5PDA _
- MIN(COLL_BAL - #FloorOCTotal, #M6TargPct * COLL_BAL)
calculate : #ClassM6PDA = MAX( 0.0, MIN(BBAL("M6"), #ClassM6PDA ))
calculate : #ClassM6PDA = MAX( 0, MIN( #ClassM6PDA, #DistribAmt - #ClassSNRPDA -
#ClassM1PDA - #ClassM2PDA - #ClassM3PDA - #ClassM4PDA - #ClassM5PDA ) )
!
!
calculate : #ClassBPDA = IF (#TrigEvent OR (#StepDown EQ 0.0)) _
THEN #DistribAmt - #ClassSNRPDA - #ClassM1PDA - #ClassM2PDA -
#ClassM3PDA - #ClassM4PDA - #ClassM5PDA - #ClassM6PDA _
ELSE BBAL("1A1", "1A2", "2A1", "2A2", "M1", "M2", "M3", "M4", "M5", "M6", "B")
- #ClassSNRPDA - #ClassM1PDA - #ClassM2PDA - #ClassM3PDA - #ClassM4PDA - #ClassM5PDA -
#ClassM6PDA _
- MIN(COLL_BAL - #FloorOCTotal, #BTargPct * COLL_BAL)
calculate : #ClassBPDA = MAX( 0.0, MIN(BBAL("B"), #ClassBPDA ))
calculate : #ClassBPDA = MAX( 0, MIN( #ClassBPDA, #DistribAmt - #ClassSNRPDA -
#ClassM1PDA - #ClassM2PDA - #ClassM3PDA - #ClassM4PDA - #ClassM5PDA - #ClassM6PDA ) )
!
!
calculate : "SNR" _
NO_CHECK CUSTOM AMOUNT = #ClassSNRPDA
!
calculate : "M1" _
NO_CHECK CUSTOM AMOUNT = #ClassM1PDA
!
calculate : "M2" _
NO_CHECK CUSTOM AMOUNT = #ClassM2PDA
!
calculate : "M3" _
NO_CHECK CUSTOM AMOUNT = #ClassM3PDA
!
calculate : "M4" _
NO_CHECK CUSTOM AMOUNT = #ClassM4PDA
!
calculate : "M5" _
NO_CHECK CUSTOM AMOUNT = #ClassM5PDA
!
calculate : "M6" _
NO_CHECK CUSTOM AMOUNT = #ClassM6PDA
!
calculate : "B" _

```

```

NO_CHECK_CUSTOM_AMOUNT      = #ClassBPDA
!
calculate : "RESID"
NO_CHECK_CUSTOM_AMOUNT      = MAX(0, #Princ - OPTIMAL_PRINCPMT("SNR", "M1", "M2",
"M3", "M4", "M5", "M6", "B"))
!
-----
pay : CLASS INTEREST PRO_RATA ( "SNR" )
-----
{#SNR_Int}
-----
pay : CLASS INTEREST PRO_RATA ( "M1" )
pay : CLASS INTEREST PRO_RATA ( "M2" )
pay : CLASS INTEREST PRO_RATA ( "M3" )
pay : CLASS INTEREST PRO_RATA ( "M4" )
pay : CLASS INTEREST PRO_RATA ( "M5" )
pay : CLASS INTEREST PRO_RATA ( "M6" )
pay : CLASS INTEREST PRO_RATA ( "B" )
pay : CLASS INTSHORT PRO_RATA ( "SNR" )
-----
{#SNR_InS}
-----
pay : CLASS INTSHORT PRO_RATA ( "M1" )
pay : CLASS INTSHORT PRO_RATA ( "M2" )
pay : CLASS INTSHORT PRO_RATA ( "M3" )
pay : CLASS INTSHORT PRO_RATA ( "M4" )
pay : CLASS INTSHORT PRO_RATA ( "M5" )
pay : CLASS INTSHORT PRO_RATA ( "M6" )
pay : CLASS INTSHORT PRO_RATA ( "B" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "SNR" )
-----
{#SNR_Prn}{1}
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "M1" )
-----
{#M1_Prn}
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "M2" )
-----
{#M2_Prn}
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "M3" )
-----
{#M3_Prn}
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "M4" )
-----
{#M4_Prn}
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "M5" )
-----
{#M5_Prn}
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "M6" )
-----

```

{#M6_Prn}

pay : CLASS PRINCIPAL SEQUENTIAL ("B")

{#B_Prn}

from : CLASS ("ROOT")

pay : CLASS COUPONCAP_SHORT PRO_RATA ("1A1"; "1A2"; "2A1"; "2A2"; "M1"; "M2"; "M3";
"M4"; "M5"; "M6"; "B")

from : CLASS ("ROOT")

pay : CLASS PRINCSHORT_LOSS SEQUENTIAL ("M1")

from : CLASS ("ROOT")

pay : CLASS PRINCSHORT_LOSS SEQUENTIAL ("M2")

from : CLASS ("ROOT")

pay : CLASS PRINCSHORT_LOSS SEQUENTIAL ("M3")

from : CLASS ("ROOT")

pay : CLASS PRINCSHORT_LOSS SEQUENTIAL ("M4")

from : CLASS ("ROOT")

pay : CLASS PRINCSHORT_LOSS SEQUENTIAL ("M5")

from : CLASS ("ROOT")

pay : CLASS PRINCSHORT_LOSS SEQUENTIAL ("M6")

from : CLASS ("ROOT")

pay : CLASS PRINCSHORT_LOSS SEQUENTIAL ("B")

pay : CLASS INTEREST PRO_RATA ("X")

pay : CLASS INTSHORT PRO_RATA ("X")

from : HEDGE ("CAP")

subject to : CEILING ((HEDGE ("CAP", "OPTIMAL_PMT") - HEDGE ("CAP", "ACTUAL_PMT")))

pay : CLASS COUPONCAP_SHORT PRO_RATA ("1A1"; "1A2"; "2A1"; "2A2"; "M1"; "M2"; "M3";
"M4"; "M5"; "M6"; "B")

from : HEDGE ("CAP")

pay : AS_INTEREST ("R#1")

!

from : CLASS ("ROOT")

pay : CLASS PRINCIPAL SEQUENTIAL ("RESID")

pay : AS_INTEREST ("OC#1")

pay : SEQUENTIAL ("OC#1")

calculate : #WriteDown = MAX(0.0,
BBAL("1A1#1","1A2#1","2A1#1","2A2#1","M1#1","M2#1","M3#1","M4#1","M5#1","M6#1","B#1","OC#1") -
COLL_BAL)

from : SUBACCOUNT (#Writedown)

pay : WRITEDOWN PRO_RATA ("OC#1")

from : SUBACCOUNT (#Writedown)
pay : WRITEDOWN SEQUENTIAL ("B#1")

from : SUBACCOUNT (#Writedown)
pay : WRITEDOWN SEQUENTIAL ("M6#1")

from : SUBACCOUNT (#Writedown)
pay : WRITEDOWN SEQUENTIAL ("M5#1")

from : SUBACCOUNT (#Writedown)
pay : WRITEDOWN SEQUENTIAL ("M4#1")

from : SUBACCOUNT (#Writedown)
pay : WRITEDOWN SEQUENTIAL ("M3#1")

from : SUBACCOUNT (#Writedown)
pay : WRITEDOWN SEQUENTIAL ("M2#1")

from : SUBACCOUNT (#Writedown)
pay : WRITEDOWN SEQUENTIAL ("M1#1")

calculate : #BondBal =
BBAL("1A1#1","1A2#1","2A1#1","2A2#1","M1#1","M2#1","M3#1","M4#1","M5#1","M6#1","B#1")
calculate : #BondBal1 = BBAL("SNR_1")
calculate : #BondBal2 = BBAL("SNR_2")
calculate : #OC = MAX(0, COLL_BAL - #BondBal)
calculate : #IncrOC = MAX(0, #OC - BBAL("OC#1"))

from : SUBACCOUNT (#IncrOC)
pay : WRITEUP SEQUENTIAL ("OC#1")

Schedule "AIO1" GROUP 1

DECLARE

VALUES OK

20040125	422,787,182.71
20040225	422,787,182.71
20040325	422,787,182.71
20040425	422,787,182.71
20040525	422,787,182.71
20040625	422,787,182.71
20040725	317,090,387.04
20040825	317,090,387.04
20040925	317,090,387.04
20041025	317,090,387.04
20041125	317,090,387.04
20041225	317,090,387.04
20050125	264,241,989.20
20050225	264,241,989.20
20050325	264,241,989.20
20050425	264,241,989.20
20050525	264,241,989.20
20050625	264,241,989.20

!

Schedule "AIO2" GROUP 2

DECLARE

VALUES OK

20040125	175,544,706.37
20040225	175,544,706.37
20040325	175,544,706.37
20040425	175,544,706.37
20040525	175,544,706.37
20040625	175,544,706.37
20040725	131,658,529.78
20040825	131,658,529.78
20040925	131,658,529.78
20041025	131,658,529.78
20041125	131,658,529.78
20041225	131,658,529.78
20050125	109,715,441.48
20050225	109,715,441.48
20050325	109,715,441.48
20050425	109,715,441.48
20050525	109,715,441.48
20050625	109,715,441.48

!

Schedule "AIO1_BAL" GROUP 1

DECLARE

VALUES OK

20040125	422,787,182.71
20040225	422,787,182.71
20040325	422,787,182.71
20040425	422,787,182.71
20040525	422,787,182.71
20040625	317,090,387.04
20040725	317,090,387.04
20040825	317,090,387.04
20040925	317,090,387.04
20041025	317,090,387.04
20041125	317,090,387.04
20041225	264,241,989.20
20050125	264,241,989.20
20050225	264,241,989.20
20050325	264,241,989.20
20050425	264,241,989.20
20050525	264,241,989.20

!

Schedule "AIO2_BAL" GROUP 2

DECLARE

VALUES OK

20040125	175,544,706.37
20040225	175,544,706.37
20040325	175,544,706.37
20040425	175,544,706.37
20040525	175,544,706.37
20040625	131,658,529.78
20040725	131,658,529.78
20040825	131,658,529.78
20040925	131,658,529.78
20041025	131,658,529.78
20041125	131,658,529.78

20041225	109,715,441.48
20050125	109,715,441.48
20050225	109,715,441.48
20050325	109,715,441.48
20050425	109,715,441.48
20050525	109,715,441.48

!
!
!

DEFINE TABLE "SRFTBL16" (4, 2) = "PERIOD" "SRF"

10.1	2.35
30.1	2.45
48.1	2.70
999	2.85

!

DEFINE DYNAMIC #SRF16 = LOOKUP_TBL("STEP", LOAN("AGE"), "SRFTBL16", "PERIOD",
"SRF")

!

DEFINE TABLE "SRFTBL17" (4, 2) = "PERIOD" "SRF"

10.1	0.30
30.1	0.40
48.1	0.65
999	0.80

!

DEFINE DYNAMIC #SRF17 = LOOKUP_TBL("STEP", MONTHS_DIFF(CURDATE,
DEAL_FIRST_PAYDATE) + 1, "SRFTBL17", "PERIOD", "SRF")

!

DEFINE TABLE "SRFTBL18" (4, 2) = "PERIOD" "SRF"

10.1	0.3
30.1	0.4
48.1	0.65
999	0.8

!

DEFINE DYNAMIC #SRF18 = LOOKUP_TBL("STEP", MONTHS_DIFF(CURDATE,
DEAL_FIRST_PAYDATE) + 1, "SRFTBL18", "PERIOD", "SRF")

!

DEFINE TABLE "SRFTBL19" (4, 2) = "PERIOD" "SRF"

10.1	0.3
30.1	0.4
48.1	0.65
999	0.8

!

DEFINE DYNAMIC #SRF19 = LOOKUP_TBL("STEP", MONTHS_DIFF(CURDATE,
DEAL_FIRST_PAYDATE) + 1, "SRFTBL19", "PERIOD", "SRF")

!

DEFINE TABLE "SRFTBL33" (4, 2) = "PERIOD" "SRF"

10.1	0.55644
30.1	0.65644
48.1	0.90644
999	1.05644

!

DEFINE DYNAMIC #SRF33 = LOOKUP_TBL("STEP", MONTHS_DIFF(CURDATE,
DEAL_FIRST_PAYDATE) + 1, "SRFTBL33", "PERIOD", "SRF")

!

DEFINE TABLE "SRFTBL34" (4, 2) = "PERIOD" "SRF"

10.1 0.3
30.1 0.4
48.1 0.65

999 0.8

!
DEFINE DYNAMIC #SRF34 = LOOKUP_TBL("STEP", MONTHS_DIFF(CURDATE,
DEAL_FIRST_PAYDATE) + 1, "SRFTBL34", "PERIOD", "SRF")
!

Collateral OVER

!
! Factor --Delay--
! Type Date P/Y BV Use BV for 0
! WL 20031201 9999 9999 FALSE
!

Pool#	Type	Gross	Current	Original	--Fee--	Maturity	Orig	ARM	Gross	#mos	#mos
P#mos	P#mos	Life	Reset	Life	Max	Look					
		Coupon	Factor	Balance	P/Y	BV	P/Y	BV	Term	Index	Margin
ToRst	RstPer	Cap	Cap	Floor	Negam	Back					
!! BEGINNING OF COLLATERAL											
M 1	"FREDDIE// //1 YR Arm // // 0"						WL 00	WAC	3 (	187370.22 /	
189000.00 ;	189000.00		0.5	0.5			355:5	355:5	360	NO_CHECK	ARM
LIBOR_6MO	2.75	8	12	SYNC_INT			9	2	2.75	0	45
GROUP "1A" TEASER											
M 2	"FREDDIE// //15 YR Fixed // // 0"						WL 00	WAC	7.51439 (		
31599670.85 /	31723673.00 ;	31723673.00					0.6965	0.6965	176:2	176:2	
178	NO_CHECK										
GROUP "1F"											
M 3	"FREDDIE// //2/28 Libor // // 0"						WL 00	WAC	7.55642 (		
920285880.27 /	921065225.76 ;	921065225.76					1.06684	1.06684	358:2		
358:2	360	NO_CHECK	ARM	LIBOR_6MO			6.289	23	6	SYNC_INT	14.29
1.004	7.5	0	45	INIT_PERCAP	2.333						GROUP
"1A" TEASER											
M 4	"FREDDIE// //2/28 Libor //IO// 24"						WL 00	WAC	6.58913 (		
12513514.63 /	12514524.00 ;	12514524.00					0.75562	0.75562	378:3	378:3	
384	NO_CHECK	ARM	LIBOR_6MO	2.825	22	6	SYNC_INT		12.61		1
3.28	0	45	INIT_PERCAP	2.983	BALLOON	SCHED_BOTH			360	AMORT	NONE FOR
24 GROUP "1A" TEASER											
M 5	"FREDDIE// //2/28 Libor //IO// 60"						WL 00	WAC	6.75694 (		
44052584.84 /	44054353.00 ;	44054353.00					0.85485	0.85485	416:2	416:2	
420	NO_CHECK	ARM	LIBOR_6MO	5.663	22	6	SYNC_INT		12.81		1
6.74	0	45	INIT_PERCAP	2.941	BALLOON	SCHED_BOTH			360	AMORT	NONE FOR
60 GROUP "1A" TEASER											
M 6	"FREDDIE// //30 YR Fixed // // 0"						WL 00	WAC	7.55324 (		
448968773.43 /	449488710.80 ;	449488710.80					0.9003	0.9003	351:2	351:2	
353	NO_CHECK										
GROUP "1F"											
M 7	"FREDDIE// //30 YR Fixed //IO// 60"						WL 00	WAC	7.09029 (		
3710588.70 /	3714162.00 ;	3714162.00					0.66987	0.66987	416:2	416:2	
420	NO_CHECK										
BALLOON SCHED_BOTH 360 AMORT NONE FOR 60 GROUP "1F"											
M 8	"FREDDIE// //30 YR Fixed //IO// 120"						WL 00	WAC	6.5558 (		
681400.00 /	681400.00 ;	681400.00					1.02779	1.02779	476:2	476:2	
480	NO_CHECK										
BALLOON SCHED_BOTH 360 AMORT NONE FOR 120 GROUP "1F"											

M 9 "FREDDIE// //3/27 Libor // // 0" WL 00 WAC 7.50516 (
267826352.97 / 268005149.00); 268005149.00 1.05924 1.05924 358:2
358:2 360 NO_CHECK ARM LIBOR_6MO 6.465 35 6 SYNC_INT 14.43
1.031 7.49 0 45 INIT_PERCAP 2.046 GROUP
"1A" TEASER
M 10 "FREDDIE// //3/27 Libor //IO// 60" WL 00 WAC 6.69369 (
6417517.39 / 6417920.00); 6417920.00 0.97034 0.97034 418:1 418:1
420 NO_CHECK ARM LIBOR_6MO 5.225 35 6 SYNC_INT 12.72 1.626
5.65 0 45 INIT_PERCAP 2.307 BALLOON SCHED_BOTH 360 AMORT NONE FOR
60 GROUP "1A" TEASER
M 11 "FREDDIE// //3/27 Libor //IO//120" WL 00 WAC 6.90322 (
886000.00 / 886000.00); 886000.00 0.5 0.5 478:1 478:1 480
NO_CHECK ARM LIBOR_6MO 4.272 35 6 SYNC_INT 12.54 1.636
4.8 0 45 INIT_PERCAP 2.364 BALLOON SCHED_BOTH 360 AMORT NONE FOR
120 GROUP "1A" TEASER
M 12 "FREDDIE// //5/25 Libor // // 0" WL 00 WAC 6.67753 (
1635617.67 / 1637037.00); 1637037.00 0.98497 0.98497 358:2 358:2
360 NO_CHECK ARM LIBOR_6MO 5.589 59 6 SYNC_INT 12.68 1.22
6.68 0 45 INIT_PERCAP 2.78 GROUP "1A" TEASER
M 13 "FREDDIE// //5/25 Libor //IO// 60" WL 00 WAC 6.8644 (
902250.00 / 902250.00); 902250.00 0.5 0.5 418:1 418:1 420
NO_CHECK ARM LIBOR_6MO 5.48 59 6 SYNC_INT 13.17 1
5.79 0 45 INIT_PERCAP 3.69 BALLOON SCHED_BOTH 360 AMORT NONE FOR
60 GROUP "1A" TEASER
M 14 "FREDDIE// //5/25 Libor //IO//120" WL 00 WAC 6.69631 (
565700.00 / 565700.00); 565700.00 0.5 0.5 476:2 476:2 480
NO_CHECK ARM LIBOR_6MO 6.696 59 6 SYNC_INT 13.7 1
6.7 0 45 INIT_PERCAP 3 BALLOON SCHED_BOTH 360 AMORT NONE FOR
120 GROUP "1A" TEASER
M 15 "FREDDIE// //Balloon // // 0" WL 00 WAC 9.38651 (
16861819.63 / 16886866.00); 16886866.00 0.66952 0.66952 357:3 357:3
360 NO_CHECK
BALLOON SCHED_BOTH 180 GROUP "1F"
M 16 "FREDDIE//OOMC_STEP//15/30 Libor // // 0" WL 00 WAC 8.09852 (
195962.89 / 196300.00); 196300.00 (#SRF16); 0.00 357:3 357:3
360 NO_CHECK ARM LIBOR_6MO 6.468 178 6 SYNC_INT 14.1 1
8.1 0 45 INIT_PERCAP 3 GROUP "1A" TEASER
M 17 "FREDDIE//OOMC_STEP//2/28 Libor // // 0" WL 00 WAC 7.8531 (
2994506.90 / 3000750.00); 3000750.00 (#SRF17); 0.00 356:4 356:4
360 NO_CHECK ARM LIBOR_6MO 5.481 21 6 SYNC_INT 13.85 1
7.85 0 45 INIT_PERCAP 3 GROUP "1A" TEASER
M 18 "FREDDIE//OOMC_STEP//30 YR Fixed // // 0" WL 00 WAC 7.49026 (
1138993.66 / 1143175.00); 1143175.00 (#SRF18); 0.00 356:4 356:4
360 NO_CHECK
GROUP "1F"
M 19 "FREDDIE//OOMC_STEP//3/27 Libor // // 0" WL 00 WAC 8.50949 (
188757.26 / 189000.00); 189000.00 (#SRF19); 0.00 357:3 357:3
360 NO_CHECK ARM LIBOR_6MO 5.467 34 6 SYNC_INT 14.51 1
8.51 0 45 INIT_PERCAP 3 GROUP "1A" TEASER
M 20 "NONCONF// //15 YR Fixed // // 0" WL 00 WAC 7.19399 (
6647510.51 / 6714050.00); 6714050.00 0.66265 0.66265 172:3 172:3
175 NO_CHECK
GROUP "2F"
M 21 "NONCONF// //2/28 Libor // // 0" WL 00 WAC 7.29881 (
364690380.70 / 365003528.50); 365003528.50 1.02112 1.02112 358:2

358:2 360 NO_CHECK ARM LIBOR_6MO 6.113 23 6 SYNC_INT 13.79
 1.357 7.14 0 45 INIT_PERCAP 2.512 GROUP
 "2A" TEASER
 M 22 "NONCONF// //2/28 Libor //IO// 24" WL 00 WAC 6.25241 (
 16044405.47 / 16046115.00); 16046115.00 0.72799 0.72799 380:2 380:2
 384 NO_CHECK ARM LIBOR_6MO 2.829 22 6 SYNC_INT 12.27 1
 3.93 0 45 INIT_PERCAP 2.979 BALLOON SCHED_BOTH 360 AMORT NONE FOR
 24 GROUP "2A" TEASER
 M 23 "NONCONF// //2/28 Libor //IO// 60" WL 00 WAC 6.4781 (
 28676165.89 / 28676248.00); 28676248.00 0.89771 0.89771 416:2 416:2
 420 NO_CHECK ARM LIBOR_6MO 5.506 23 6 SYNC_INT 12.5 1.012
 6.45 0 45 INIT_PERCAP 2.974 BALLOON SCHED_BOTH 360 AMORT NONE FOR
 60 GROUP "2A" TEASER
 M 24 "NONCONF// //2/28 Libor //IO//120" WL 00 WAC 7.125 (
 352000.00 / 352000.00); 352000.00 0.5 0.5 474:3 474:3 480
 NO_CHECK ARM LIBOR_6MO 4.625 22 6 SYNC_INT 12.13 1
 4.63 0 45 INIT_PERCAP 3 BALLOON SCHED_BOTH 360 AMORT NONE FOR
 120 GROUP "2A" TEASER
 M 25 "NONCONF// //30 YR Fixed // // 0" WL 00 WAC 7.22092 (
 201122840.22 / 201740328.80); 201740328.80 0.89518 0.89518 354:3
 354:3 357 NO_CHECK
 GROUP "2F"
 M 26 "NONCONF// //30 YR Fixed //IO// 60" WL 00 WAC 6.62737 (
 4385657.21 / 4385950.00); 4385950.00 0.95984 0.95984 416:2 416:2
 420 NO_CHECK
 BALLOON SCHED_BOTH 360 AMORT NONE FOR 60 GROUP "2F"
 M 27 "NONCONF// //30 YR Fixed //IO//120" WL 00 WAC 7.0475 (
 20709605.30 / 20710100.00); 20710100.00 0.5 0.5 478:1 478:1
 480 NO_CHECK
 BALLOON SCHED_BOTH 360 AMORT NONE FOR 120 GROUP "2F"
 M 28 "NONCONF// //3/27 Libor // // 0" WL 00 WAC 7.1847 (
 64101599.09 / 64155448.88); 64155448.88 0.97948 0.97948 358:2 358:2
 360 NO_CHECK ARM LIBOR_6MO 6.323 35 6 SYNC_INT 14.05 1.097
 7.15 0 45 INIT_PERCAP 2.071 GROUP "2A" TEASER
 M 29 "NONCONF// //3/27 Libor //IO// 60" WL 00 WAC 6.60235 (
 3912420.00 / 3912420.00); 3912420.00 0.64615 0.64615 416:2 416:2
 420 NO_CHECK ARM LIBOR_6MO 5.065 35 6 SYNC_INT 12.7 1.716
 5.61 0 45 INIT_PERCAP 2.188 BALLOON SCHED_BOTH 360 AMORT NONE FOR
 60 GROUP "2A" TEASER
 M 30 "NONCONF// //5/25 Libor // // 0" WL 00 WAC 6.24913 (
 1353781.89 / 1356400.00); 1356400.00 0.5 0.5 358:2 358:2 360
 NO_CHECK ARM LIBOR_6MO 5.911 58 6 SYNC_INT 12.25 1
 6.25 0 45 INIT_PERCAP 3 GROUP "2A" TEASER
 M 31 "NONCONF// //5/25 Libor //IO// 60" WL 00 WAC 6.74232 (
 1015200.00 / 1015200.00); 1015200.00 0.5 0.5 418:1 418:1 420
 NO_CHECK ARM LIBOR_6MO 6.742 60 6 SYNC_INT 13.74 1
 6.74 0 45 INIT_PERCAP 3 BALLOON SCHED_BOTH 360 AMORT NONE FOR
 60 GROUP "2A" TEASER
 M 32 "NONCONF// //Balloon // // 0" WL 00 WAC 8.52993 (
 16937252.43 / 17135786.00); 17135786.00 0.85696 0.85696 355:5 355:5
 360 NO_CHECK
 BALLOON SCHED_BOTH 180 GROUP "2F"
 M 33 "NONCONF//OOMC_STEP//2/28 Libor // // 0" WL 00 WAC 7.92814 (
 1330074.39 / 1332650.00); 1332650.00 (#SRF33); 0.00 356:4 356:4
 360 NO_CHECK ARM LIBOR_6MO 5.314 21 6 SYNC_INT 13.93 1
 7.93 0 45 INIT_PERCAP 3 GROUP "2A" TEASER

M 34 "NONCONF//OOMC_STEP//30 YR Fixed // 0" WL 00 WAC 7.23576 (
157383.45 / 157700.00); 157700.00 (#SRF34); 0.00 357:3 357:3
360 NO_CHECK
GROUP "2F"