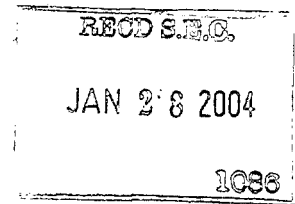


UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
WASHINGTON, DC 20549



FORM SE

FORM FOR SUBMISSION OF PAPER FORMAT EXHIBITS
BY ELECTRONIC FILERS

Merrill Lynch Mortgage Investors, Inc.
(Exact Name of Registrant as Specified in Charter)

0000809940
(Registrant CIK Number)

Form 8-K for January 26, 2004
(Electronic Report, Schedule or Registration Statement of
Which the Documents Are a Part (Give Period of Report))

333-106175
(SEC File Number, if Available)

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

PROCESSED
JAN 30 2004
THOMSON
FINANCIAL

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 January 26, 2004.

MERRILL LYNCH MORTGAGE INVESTORS, INC.

By: Ted Bouloukos
Name: Ted Bouloukos
Title: Vice President

Exhibit Index

<u>Exhibit</u>	<u>Page</u>
99.1 Computational Materials	5
99.2 Computational Materials	35

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

COMPUTATIONAL MATERIALS

for

FIELDSTONE MORTGAGE INVESTMENT TRUST, SERIES 2004-1

! FIELDSTONE_04_1_CM_5_1.CDI #CMOVER_3.0D ASSET_BACKED_HOMEEQUITY !
MAX_CF_VECTSIZE 552

!
!! Created by Intex Deal Maker v3.5.327 , subroutines 3.0f
!! 01/20/2004 8:46 AM

!
! Modeled in the Intex CMO Modeling Language, (DNY10ABF21)
! which is copyright (c) 2003 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 3
GROUP 2 = "2A" "2B"
GROUP 3 = "3A" "3B"

!
! DEFINE CONSTANT #OrigCollBal = 680857782.45
! DEFINE CONSTANT #OrigCollBal1 = 0.02
! DEFINE CONSTANT #OrigCollBal2 = 540335968.48
! DEFINE CONSTANT #OrigCollBal3 = 140521813.95

! DEFINE CONSTANT #OrigBondBal = 663157000.02
! DEFINE CONSTANT #OrigBondBal1 = 0.02
! DEFINE CONSTANT #OrigBondBal2 = 540335968.48
! DEFINE CONSTANT #OrigBondBal3 = 140521813.95

! DEFINE CONSTANT #SpecSenEnhPct = 38%
! DEFINE CONSTANT #AAATargPct = 62%
! DEFINE CONSTANT #AATargPct = 74.3%
! DEFINE CONSTANT #ATargPct = 83.8%
! DEFINE CONSTANT #AMTargPct = 86.3%
! DEFINE CONSTANT #BBBPTargPct = 89.8%
! DEFINE CONSTANT #BBBTargPct = 91.8%
! DEFINE CONSTANT #BBBMTargPct = 94.8%
! DEFINE #BondBal = 663157000.02

! FULL_DEALNAME: FINAL COMPMAT FILE TIED OUT WITH KPMG **2 Groups**

! DEAL SIZE: \$ 663157000.02
! PRICING SPEED: GROUP 1 CPR 28%
! PRICING SPEED: GROUP "2A" CPR 28%
! PRICING SPEED: GROUP "2B" CPR 28%
! PRICING SPEED: GROUP "3A" CPR 28%
! PRICING SPEED: GROUP "3B" CPR 28%

! ISSUE DATE: 20040101
! SETTLEMENT DATE: 20040210

! Record date delay: 24

! DEFINE TR_INDEXDEPS_ALL

! DEFINE SCHEDULE "Fieldstone04-1CM4b","NAS_IO_BAL"

! DEFINE DYNAMIC #PayMonth = MONTH(CURDATE)
! DEFINE DYNAMIC #FeePayDate = 1

```

!
DEFINE DYNAMIC #TrusteeAmt = IF #FeePayDate _
    THEN MIN( COLL_I, 250 ) _
    ELSE 0
!
TRUSTEE_FEE 0.0 ADDL_AMT #TrusteeAmt
!
DEAL_CLOCK_INFO _
ISSUE_CDU_DATE 20040101 _
DEAL_FIRSTPAY_DATE 20040225
!
!
DEFINE TABLE "CapNotional" (25, 2) = "CURDATE" "Balance"
20040225.1 312028820.00
20040325.1 310406035.00
20040425.1 308278857.00
20040525.1 305647611.00
20040625.1 302514797.00
20040725.1 298885851.00
20040825.1 294768378.00
20040925.1 290164669.00
20041025.1 285087341.00
20041125.1 279551561.00
20041225.1 273579330.00
20050125.1 267190846.00
20050225.1 260408518.00
20050325.1 253665965.00
20050425.1 246967436.00
20050525.1 240317801.00
20050625.1 233935261.00
20050725.1 227813387.00
20050825.1 221935275.00
20050925.1 216285945.00
20051025.1 210849362.00
20051125.1 205418164.00
20051225.1 197542646.00
20060125.1 186995760.00
20060225.1 0
!
DEFINE DYNAMIC #CapBal = LOOKUP_TBL( "STEP", Curdate, "CapNotional", "CURDATE", "Balance"
)
!
DEFINE DYNAMIC #CapBalEnd = LOOKUP_TBL( "STEP", Curdate + 30, "CapNotional", "CURDATE",
"Balance" )
!
!
DEFINE TABLE "SwapNotional" (25, 2) = "CURDATE" "Balance"
20040225.1 312028820.00
20040325.1 310406035.00
20040425.1 308278857.00
20040525.1 305647611.00
20040625.1 302514797.00
20040725.1 298885851.00
20040825.1 294768378.00
20040925.1 290164669.00

```

20041025.1	285087341.00
20041125.1	279551561.00
20041225.1	273579330.00
20050125.1	267190846.00
20050225.1	260408518.00
20050325.1	253665965.00
20050425.1	246967436.00
20050525.1	240317801.00
20050625.1	233935261.00
20050725.1	227813387.00
20050825.1	221935275.00
20050925.1	216285945.00
20051025.1	210849362.00
20051125.1	205418164.00
20051225.1	197542646.00
20060125.1	186995760.00
20060225.1	0

!
 DEFINE DYNAMIC #SwapBal = LOOKUP_TBL("STEP", Curdate, "SwapNotional", "CURDATE",
 "Balance")

!
 DEFINE DYNAMIC #SwapBalEnd = LOOKUP_TBL("STEP", Curdate + 30, "SwapNotional",
 "CURDATE", "Balance")

!
 !
 DEFINE TABLE "Cap2Notional" (25, 2) = "CURDATE" "Balance"

20040225.1	100000000
20040325.1	99479924.64
20040425.1	98798199.79
20040525.1	97954929.61
20040625.1	96950915.3
20040725.1	95787899.02
20040825.1	94468318.02
20040925.1	92992906.55
20041025.1	91365708.14
20041125.1	89591583.56
20041225.1	87677583.76
20050125.1	85630181.85
20050225.1	83456559.56
20050325.1	81295684.48
20050425.1	79148918.36
20050525.1	77017821.94
20050625.1	74972324.99
20050725.1	73010367.12
20050825.1	71126530.88
20050925.1	68316015.42
20051025.1	65573681.82
20051125.1	62833074.01
20051225.1	59309102.67
20060125.1	0
20060225.1	0

!
 DEFINE DYNAMIC #CapBal2 = LOOKUP_TBL("STEP", Curdate, "Cap2Notional", "CURDATE",
 "Balance")

!

```

DEFINE DYNAMIC #CapBal2End = LOOKUP_TBL( "STEP", Curdate + 30, "Cap2Notional",
"CURDATE", "Balance" )

```

```

!
!

```

```

DEFINE TABLE "Swap2Notional" (25, 2) = "CURDATE" "Balance"

```

20040225.1	100000000
20040325.1	99479924.64
20040425.1	98798199.79
20040525.1	97954929.61
20040625.1	96950915.3
20040725.1	95787899.02
20040825.1	94468318.02
20040925.1	92992906.55
20041025.1	91365708.14
20041125.1	89591583.56
20041225.1	87677583.76
20050125.1	85630181.85
20050225.1	83456559.56
20050325.1	81295684.48
20050425.1	79148918.36
20050525.1	77017821.94
20050625.1	74972324.99
20050725.1	73010367.12
20050825.1	71126530.88
20050925.1	68316015.42
20051025.1	65573681.82
20051125.1	62833074.01
20051225.1	59309102.67
20060125.1	0
20060225.1	0

```

!

```

```

DEFINE DYNAMIC #SwapBal2 = LOOKUP_TBL( "STEP", Curdate, "Swap2Notional", "CURDATE",
"Balance" )

```

```

!

```

```

DEFINE DYNAMIC #SwapBal2End = LOOKUP_TBL( "STEP", Curdate + 30, "Swap2Notional",
"CURDATE", "Balance" )

```

```

!

```

```

!

```

```

DEFINE TABLE "Cap3Notional" (24, 2) = "CURDATE" "Balance"

```

20040225.1	241069679.75
20040325.1	238412437.03
20040425.1	235262357.78
20040525.1	231625264.75
20040625.1	227510262.43
20040725.1	222924958.29
20040825.1	217877571.37
20040925.1	212392556.10
20041025.1	206496031.34
20041125.1	200217478.31
20041225.1	193579393.33
20050125.1	186842529.32
20050225.1	180417259.66
20050325.1	174079000.73
20050425.1	167961787.23
20050525.1	162268681.48

```

20050625.1 156855935.90
20050725.1 151695104.82
20050825.1 146760230.39
20050925.1 142770986.96
20051025.1 137292415.14
20051125.1 107922161.26
20051225.1 0
20060125.1 0
!
DEFINE DYNAMIC #CapBal3 = LOOKUP_TBL( "STEP", Curdate, "Cap3Notional", "CURDATE",
"Balance" )
!
DEFINE DYNAMIC #CapBal3End = LOOKUP_TBL( "STEP", Curdate + 30, "Cap3Notional",
"CURDATE", "Balance" )
!
!
DEFINE TABLE "Swap3Notional" (24, 2) = "CURDATE" "Balance"
20040225.1 241069679.75
20040325.1 238412437.03
20040425.1 235262357.78
20040525.1 231625264.75
20040625.1 227510262.43
20040725.1 222924958.29
20040825.1 217877571.37
20040925.1 212392556.10
20041025.1 206496031.34
20041125.1 200217478.31
20041225.1 193579393.33
20050125.1 186842529.32
20050225.1 180417259.66
20050325.1 174079000.73
20050425.1 167961787.23
20050525.1 162268681.48
20050625.1 156855935.90
20050725.1 151695104.82
20050825.1 146760230.39
20050925.1 142770986.96
20051025.1 137292415.14
20051125.1 107922161.26
20051225.1 0
20060125.1 0
!
DEFINE DYNAMIC #SwapBal3 = LOOKUP_TBL( "STEP", Curdate, "Swap3Notional", "CURDATE",
"Balance" )
!
DEFINE DYNAMIC #SwapBal3End = LOOKUP_TBL( "STEP", Curdate + 30, "Swap3Notional",
"CURDATE", "Balance" )
!
!
DEFINE TABLE "Cap4Notional" (24, 2) = "CURDATE" "Balance"
20040225.1 27759282.68
20040325.1 27453299.98
20040425.1 27090566.93
20040525.1 26671754.02
20040625.1 26197909.65

```

20040725.1	25669909.80
20040825.1	25088700.91
20040925.1	24457098.92
20041025.1	23778111.43
20041125.1	23055133.21
20041225.1	22290754.72
20050125.1	21515001.78
20050225.1	20775129.07
20050325.1	20045275.69
20050425.1	19340875.79
20050525.1	18685312.08
20050625.1	18062031.98
20050725.1	17467759.95
20050825.1	16899506.92
20050925.1	16440143.74
20051025.1	15809283.72
20051125.1	15660461.53
20051225.1	0.00
20060125.1	0.00

!

```
DEFINE DYNAMIC #CapBal4 = LOOKUP_TBL( "STEP", Curdate, "Cap4Notional", "CURDATE",
"Balance" )
```

!

```
DEFINE DYNAMIC #CapBal4End = LOOKUP_TBL( "STEP", Curdate + 30, "Cap4Notional",
"CURDATE", "Balance" )
```

!

```
DEFINE TABLE "Swap4Notional" (24, 2) = "CURDATE" "Balance"
```

20040225.1	27759282.68
20040325.1	27453299.98
20040425.1	27090566.93
20040525.1	26671754.02
20040625.1	26197909.65
20040725.1	25669909.80
20040825.1	25088700.91
20040925.1	24457098.92
20041025.1	23778111.43
20041125.1	23055133.21
20041225.1	22290754.72
20050125.1	21515001.78
20050225.1	20775129.07
20050325.1	20045275.69
20050425.1	19340875.79
20050525.1	18685312.08
20050625.1	18062031.98
20050725.1	17467759.95
20050825.1	16899506.92
20050925.1	16440143.74
20051025.1	15809283.72
20051125.1	15660461.53
20051225.1	0.00
20060125.1	0.00

!

```
DEFINE DYNAMIC #SwapBal4 = LOOKUP_TBL( "STEP", Curdate, "Swap4Notional", "CURDATE",
"Balance" )
```

```

!
DEFINE DYNAMIC #SwapBal4End = LOOKUP_TBL( "STEP", Curdate + 30, "Swap4Notional",
"CURDATE", "Balance" )
!
!
DEFINE #FloorCollat      = .5% * #OrigCollBal
DEFINE #SDTrigDelinqFrac = 0
DEFINE #SpecOCTarg       = 2.6% * #OrigCollBal
ifndef #cmover_3.0d _
  DEFINE #OC              = 17700782.43
!
!
ifdef #cmover_3.0d _
  DEFINE STANDARDIZE OC_ACTUAL_VAL      #OC      = 17700782.43
!
  DEFINE STANDARDIZE OCT_INITVAL      CONSTANT #InitOCTarg = 2.6% * #OrigCollBal
  DEFINE STANDARDIZE OCT_STEPDOWN_MONTH CONSTANT #StepDownDate = 37
  DEFINE STANDARDIZE OCT_STEPDOWN_FRAC CONSTANT #StepOCFrac = 0.052
  DEFINE STANDARDIZE EXCESS_INTEREST   #XSSpread = 0
  DEFINE STANDARDIZE OCT_FLOOR          CONSTANT #FloorOCTarg = #FloorCollat
  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 #NetRate3 = ( COLL_I_MISC("COUPON", 3) ) / COLL_PREV_BAL(3) *
1200
!
  DEFINE DYNAMIC STICKY #NetRateActual360 = #Netrate * 30 / DAYS_DIFF(CURDATE ,
MONTHS_ADD(CURDATE,-1))
!
  DEFINE COLLAT WT_BY_PREVBAL #LifeCap = IF LOAN("LIFE_CAP") GT 0 _
      THEN LOAN("LIFE_CAP") - (LOAN("GROSSRATE") - LOAN("NETRATE")) _
      ELSE LOAN("NETRATE")
  DEFINE COLLAT WT_BY_PREVBAL GROUP 2 #LifeCap2 = IF LOAN("LIFE_CAP") GT 0 _
      THEN LOAN("LIFE_CAP") - (LOAN("GROSSRATE") - LOAN("NETRATE")) _
      ELSE LOAN("NETRATE")
  DEFINE COLLAT WT_BY_PREVBAL GROUP 3 #LifeCap3 = IF LOAN("LIFE_CAP") GT 0 _
      THEN LOAN("LIFE_CAP") - (LOAN("GROSSRATE") - LOAN("NETRATE")) _
      ELSE LOAN("NETRATE")
  DEFINE COLLAT WT_BY_PREVBAL GROUP 1 #LifeCap1 = IF LOAN("LIFE_CAP") GT 0 _
      THEN LOAN("LIFE_CAP") - (LOAN("GROSSRATE") - LOAN("NETRATE")) _
      ELSE LOAN("NETRATE")
!
  DEFINE DYNAMIC #NAS_IO_SCHED = SCHED_AMOUNT("Fieldstone04-1CM4b")
  DEFINE DYNAMIC #NAS_IO_ENDBAL = SCHED_AMOUNT("NAS_IO_BAL")
!
!
!
TOLERANCE WRITEDOWN_0LOSS 999999999999.00
TOLERANCE INTEREST      999999999999.00
!
INITIAL INDEX  LIBOR_6MO      1.21
INITIAL INDEX  LIBOR_1MO      1.11

```

```

INITIAL INDEX  CMT_1YR      1.3
!
DEFINE TRANCHE "CAP_IN", "SWAP_IN", "SWAP_OUT", "CAP_IN2", "SWAP_IN2", "SWAP_OUT2",
"CAP_IN3", "SWAP_IN3", "SWAP_OUT3", "CAP_IN4", "SWAP_IN4", "SWAP_OUT4", "NAS_IO", "1A1",
"1A2", "2A", "3A", "AA", "A", "A-", "BBB+", "BBB", "BBB-", "INV_IO", "R_PP", "NIM", "NIM_R"
!
!
DEFINE DYNAMIC #HedgeRestrictBal5 =
BBAL("1A1","1A2","2A","3A","AA","A","A-","BBB+","BBB","BBB-")
!
DEFINE DYNAMIC #HedgeCapBeginBal = MIN( #HedgeRestrictBal5, #CapBal )
!
DEFINE DYNAMIC #HedgeCapEndBal = MIN( #HedgeRestrictBal5, #CapBalEnd )
!
DEFINE DYNAMIC #HedgeRestrictBal6 =
BBAL("1A1","1A2","2A","3A","AA","A","A-","BBB+","BBB","BBB-")
!
DEFINE DYNAMIC #HedgeSwapBeginBal = MIN( #HedgeRestrictBal6, #SwapBal )
!
DEFINE DYNAMIC #HedgeSwapEndBal = MIN( #HedgeRestrictBal6, #SwapBalEnd )
!
DEFINE DYNAMIC #HedgeRestrictBal7 =
BBAL("1A1","1A2","2A","3A","AA","A","A-","BBB+","BBB","BBB-")
!
DEFINE DYNAMIC #HedgeCapBeginBal2 = MIN( #HedgeRestrictBal7, #CapBal2 )
!
DEFINE DYNAMIC #HedgeCapEndBal2 = MIN( #HedgeRestrictBal7, #CapBal2End )
!
DEFINE DYNAMIC #HedgeRestrictBal8 =
BBAL("1A1","1A2","2A","3A","AA","A","A-","BBB+","BBB","BBB-")
!
DEFINE DYNAMIC #HedgeSwapBeginBal2 = MIN( #HedgeRestrictBal8, #SwapBal2 )
!
DEFINE DYNAMIC #HedgeSwapEndBal2 = MIN( #HedgeRestrictBal8, #SwapBal2End )
!
DEFINE DYNAMIC #HedgeRestrictBal9 =
BBAL("1A1","1A2","2A","3A","AA","A","A-","BBB+","BBB","BBB-")
!
DEFINE DYNAMIC #HedgeCapBeginBal3 = MIN( #HedgeRestrictBal9, #CapBal3 )
!
DEFINE DYNAMIC #HedgeCapEndBal3 = MIN( #HedgeRestrictBal9, #CapBal3End )
!
DEFINE DYNAMIC #HedgeRestrictBal10 =
BBAL("1A1","1A2","2A","3A","AA","A","A-","BBB+","BBB","BBB-")
!
DEFINE DYNAMIC #HedgeSwapBeginBal3 = MIN( #HedgeRestrictBal10, #SwapBal3 )
!
DEFINE DYNAMIC #HedgeSwapEndBal3 = MIN( #HedgeRestrictBal10, #SwapBal3End )
!
DEFINE DYNAMIC #HedgeRestrictBal11 =
BBAL("1A1","1A2","2A","3A","AA","A","A-","BBB+","BBB","BBB-")
!
DEFINE DYNAMIC #HedgeCapBeginBal4 = MIN( #HedgeRestrictBal11, #CapBal4 )
!
DEFINE DYNAMIC #HedgeCapEndBal4 = MIN( #HedgeRestrictBal11, #CapBal4End )

```

```

!
DEFINE DYNAMIC #HedgeRestrictBal12 =
BBAL("1A1","1A2","2A","3A","AA","A","A-","BBB+","BBB","BBB-")
!
DEFINE DYNAMIC #HedgeSwapBeginBal4 = MIN( #HedgeRestrictBal12, #SwapBal4 )
!
DEFINE DYNAMIC #HedgeSwapEndBal4 = MIN( #HedgeRestrictBal12, #SwapBal4End )
!
Tranche "CAP_IN" PSEUDO HEDGE
  Block $ 312028820.00 at 1.11 FLOAT NOTIONAL WITH FORMULA BEGIN ( #HedgeCapBeginBal );
END ( #HedgeCapEndBal ); _
  DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
  Delay 0 Dated 20040210 Next 20040225
  ((1 * MIN(2.247, LIBOR_1MO)) + (-0))
  0 999
!
!!*****Delete "AT" Script*****
Tranche "SWAP_IN" PSEUDO HEDGE
  Block $ 312028820.00 FLOAT NOTIONAL WITH FORMULA BEGIN ( #HedgeSwapBeginBal ); END (
#HedgeSwapEndBal ); _
  DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
  Delay 0 Dated 20040210 Next 20040225
  1 * LIBOR_1MO + -2.247
  0 999
!
Tranche "SWAP_OUT" PSEUDO HEDGE
  Block $ 312028820.00 at 2.247 NOTIONAL WITH FORMULA BEGIN ( #HedgeSwapBeginBal ); END (
#HedgeSwapEndBal ); _
  DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
  Delay 0 Dated 20040210 Next 20040225
!
Tranche "CAP_IN2" PSEUDO HEDGE
  Block $ 100000000 at 1.11 FLOAT NOTIONAL WITH FORMULA BEGIN ( #HedgeCapBeginBal2 );
END ( #HedgeCapEndBal2 ); _
  DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
  Delay 0 Dated 20040210 Next 20040225
  ((1 * MIN(2.1065, LIBOR_1MO)) + (-0))
  0 999
!
!!*****Delete Script*****
Tranche "SWAP_IN2" PSEUDO HEDGE
  Block $ 100000000 FLOAT NOTIONAL WITH FORMULA BEGIN ( #HedgeSwapBeginBal2 ); END (
#HedgeSwapEndBal2 ); _
  DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
  Delay 0 Dated 20040210 Next 20040225
  1 * LIBOR_1MO + -2.1065
  0 999
!
Tranche "SWAP_OUT2" PSEUDO HEDGE
  Block $ 100000000 at 2.1065 NOTIONAL WITH FORMULA BEGIN ( #HedgeSwapBeginBal2 ); END (
#HedgeSwapEndBal2 ); _
  DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
  Delay 0 Dated 20040210 Next 20040225
!
Tranche "CAP_IN3" PSEUDO HEDGE

```

Block \$ 241069679.75 at 1.11 FLOAT NOTIONAL WITH FORMULA BEGIN (#HedgeCapBeginBal3);
 END (#HedgeCapEndBal3); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
 Delay 0 Dated 20040210 Next 20040225
 ((1 * MIN(2.1275, LIBOR_1MO)) + (-0))
 0 999

!
 !!*****Delete "AT" Script*****

Tranche "SWAP_IN3" PSEUDO HEDGE
 Block \$ 241069679.75 FLOAT NOTIONAL WITH FORMULA BEGIN (#HedgeSwapBeginBal3); END (#HedgeSwapEndBal3); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
 Delay 0 Dated 20040210 Next 20040225
 1 * LIBOR_1MO + -2.1275
 0 999

!
 Tranche "SWAP_OUT3" PSEUDO HEDGE
 Block \$ 241069679.75 at 2.1275 NOTIONAL WITH FORMULA BEGIN (#HedgeSwapBeginBal3); END (#HedgeSwapEndBal3); _
 DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
 Delay 0 Dated 20040210 Next 20040225

!
 Tranche "CAP_IN4" PSEUDO HEDGE
 Block \$ 27759282.68 at 1.11 FLOAT NOTIONAL WITH FORMULA BEGIN (#HedgeCapBeginBal4);
 END (#HedgeCapEndBal4); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
 Delay 0 Dated 20040210 Next 20040225
 ((1 * MIN(2.005, LIBOR_1MO)) + (-0))
 0 999

!
 !!*****Delete "AT" Script*****

Tranche "SWAP_IN4" PSEUDO HEDGE
 Block \$ 27759282.68 FLOAT NOTIONAL WITH FORMULA BEGIN (#HedgeSwapBeginBal4); END (#HedgeSwapEndBal4); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
 Delay 0 Dated 20040210 Next 20040225
 1 * LIBOR_1MO + -2.005
 0 999

!
 Tranche "SWAP_OUT4" PSEUDO HEDGE
 Block \$ 27759282.68 at 2.005 NOTIONAL WITH FORMULA BEGIN (#HedgeSwapBeginBal4); END (#HedgeSwapEndBal4); _
 DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
 Delay 0 Dated 20040210 Next 20040225

!
 Tranche "NAS_IO" SEN_IO ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 187235889.77 at 6 FREQ M NOTIONAL WITH FORMULA BEGIN (MIN(COLL_PREV_BAL, #NAS_IO_SCHED)); _
 END (MIN(COLL_BAL, #NAS_IO_ENDBAL)); _
 DAYCOUNT 30360 BUSINESS_DAY NONE _
 Delay 24 Dated 20040101 Next 20040225

!
 Tranche "1A1" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 0.00 at 1.43 GROUP 1 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE ((COLL_I_MISC("COUPON")) / #BondBal * 1200); _

DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20040210 Next 20040225
 (MIN((1 * LIBOR_1MO + (IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 20%) THEN 0.64 ELSE
 0.32)), #LifeCap1 * 30 / NDAYS_ACCRUE_INT("1A1#1")))
 0 10.90

!
 Tranche "1A2" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 0.00 at 1.45 GROUP 1 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE ((COLL_I_MISC("COUPON")) / #BondBal * 1200); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20040210 Next 20040225
 (MIN((1 * LIBOR_1MO + (IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 20%) THEN 0.68 ELSE
 0.34)), #LifeCap1 * 30 / NDAYS_ACCRUE_INT("1A2#1")))
 0 10.90

!
 Tranche "2A" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 437672000.00 at 1.44 GROUP 2 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE ((COLL_I_MISC("COUPON")) / #BondBal * 1200); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20040210 Next 20040225
 (MIN((1 * LIBOR_1MO + (IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 20%) THEN 0.66 ELSE
 0.33)), #LifeCap2 * 30 / NDAYS_ACCRUE_INT("2A#1")))
 0 10.65

!
 Tranche "3A" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 113823000.02 at 1.41 GROUP 3 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE ((COLL_I_MISC("COUPON")) / #BondBal * 1200); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20040210 Next 20040225
 (MIN((1 * LIBOR_1MO + (IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 20%) THEN 0.6 ELSE 0.3
)), #LifeCap3 * 30 / NDAYS_ACCRUE_INT("3A#1")))
 0 10.85

!
 Tranche "AA" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 41873000.00 at 1.76 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE ((COLL_I_MISC("COUPON")) / #BondBal * 1200); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20040210 Next 20040225
 (MIN((1 * LIBOR_1MO + (IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 20%) THEN 0.975 ELSE
 0.65)), #LifeCap * 30 / NDAYS_ACCRUE_INT("AA#1")))
 0 10.65

!
 Tranche "A" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 32341000.00 at 2.46 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE ((COLL_I_MISC("COUPON")) / #BondBal * 1200); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20040210 Next 20040225
 (MIN((1 * LIBOR_1MO + (IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 20%) THEN 2.025 ELSE
 1.35)), #LifeCap * 30 / NDAYS_ACCRUE_INT("A#1")))
 0 10.65

!
 Tranche "A-" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 8511000.00 at 2.66 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE ((COLL_I_MISC("COUPON")) / #BondBal * 1200); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _

Delay 0 Dated 20040210 Next 20040225
 (MIN((1 * LIBOR_1MO + (IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 20%) THEN 2.325 ELSE
 1.55)), #LifeCap * 30 / NDAYS_ACCRUE_INT("A-#1")))
 0 10.65

!
 Tranche "BBB+" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 11915000.00 at 2.86 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE ((COLL_I_MISC("COUPON")) / #BondBal * 1200); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20040210 Next 20040225
 (MIN((1 * LIBOR_1MO + (IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 20%) THEN 2.625 ELSE
 1.75)), #LifeCap * 30 / NDAYS_ACCRUE_INT("BBB-#1")))
 0 10.65

!
 Tranche "BBB" JUN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 6809000.00 at 3.06 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE ((COLL_I_MISC("COUPON")) / #BondBal * 1200); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20040210 Next 20040225
 (MIN((1 * LIBOR_1MO + (IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 20%) THEN 2.925 ELSE
 1.95)), #LifeCap * 30 / NDAYS_ACCRUE_INT("BBB-#1")))
 0 10.65

!
 Tranche "BBB-" JUN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 10213000.00 at 4.36 FREQ M FLOAT RESET M _
 COUPONCAP 30360 NONE ((COLL_I_MISC("COUPON")) / #BondBal * 1200); _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20040210 Next 20040225
 (MIN((1 * LIBOR_1MO + (IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 20%) THEN 4.875 ELSE
 3.25)), #LifeCap * 30 / NDAYS_ACCRUE_INT("BBB-#1")))
 0 10.65

!
 Tranche "INV_IO" SEN_INV_IO ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 663157000.02 at 0.89 FREQ M FLOAT RESET M NOTIONAL WITH BLOCK
 "1A1#1"&"1A2#1"&"2A#1"&"3A#1"&"AA#1"&"A#1"&"A-#1"&"BBB-#1"&"BBB-#1"&"BBB-#1" _
 DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
 Delay 0 Dated 20040210 Next 20040225
 (MIN(-1 * LIBOR_1MO + 2, COLL_I / BBAL("INV_IO#1") * 1200))
 0 999

!
 Tranche "R" JUN_RES
 Block 680857782.45 at 0 NOTIONAL WITH GROUP 0 SURPLUS _
 DAYCOUNT 30360 BUSINESS_DAY NONE _
 FREQ M Delay 24 Dated 20040201 Next 20040225

!
 Tranche "R_PP" JUN_PEN_NO
 Block 680857782.45 at 0 NOTIONAL WITH GROUP 0 _
 DAYCOUNT 30360 BUSINESS_DAY NONE _
 FREQ M Delay 24 Dated 20040201 Next 20040225

!
 Tranche "NIM" JUN_NIM_FIX ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 19310000.00 at 8.5 NO_SUPPORT_COUPON EXCLUDE_FROM_CREDSUPP FREQ M _
 DAYCOUNT 30360 BUSINESS_DAY NONE _
 Delay 24 Dated 20040101 Next 20040225

!

Tranche "NIM_R" JUN_RES ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
 Block 19310000.00 at 0 NOTIONAL WITH BLOCK "NIM#1" _
 DAYCOUNT 30360 BUSINESS_DAY NONE _
 FREQ M Delay 24 Dated 20040201 Next 20040225

!
 Tranche "#NetRate" SYMVAR
 !
 Tranche "#NetRateActual360" SYMVAR
 !
 Tranche "#OC" SYMVAR
 Tranche "#SpecOCTarg" SYMVAR
 !

Tranche "DEAL_PLUGIN" PSEUDO
 Block USE PCT 100.0 100.0 OF "1A1#1"
 Block USE PCT 100.0 100.0 OF "1A2#1"
 Block USE PCT 100.0 100.0 OF "2A#1"
 Block USE PCT 100.0 100.0 OF "3A#1"
 Block USE PCT 100.0 100.0 OF "AA#1"
 Block USE PCT 100.0 100.0 OF "A#1"
 Block USE PCT 100.0 100.0 OF "A-#1"
 Block USE PCT 100.0 100.0 OF "BBB+#1"
 Block USE PCT 100.0 100.0 OF "BBB#1"
 Block USE PCT 100.0 100.0 OF "BBB-#1"
 Block USE PCT 0.0 100.0 OF "R#1"
 Block USE PCT 0.0 100.0 OF "R_PP#1"
 Block USE PCT 0.0 100.0 OF "INV_IO#1"
 Block USE PCT 0.0 100.0 OF "NAS_IO#1"

!
 !
 DEFINE PSEUDO_TRANCHE COLLAT _
 Delay 24 Dated 20040101 Next 20040225 Settle 20040210
 DEFINE PSEUDO_TRANCHE COLLAT GROUP 1 _
 Delay 24 Dated 20040101 Next 20040225 Settle 20040210
 DEFINE PSEUDO_TRANCHE COLLAT GROUP 2 _
 Delay 24 Dated 20040101 Next 20040225 Settle 20040210
 DEFINE PSEUDO_TRANCHE COLLAT GROUP 3 _
 Delay 24 Dated 20040101 Next 20040225 Settle 20040210

!
 TOLERANCE TRANCHE_WRITEDOWN_0LOSS "NIM" 19310000
 !

RESERVE_FUND "YmRsvFnd" FUNDING_FROM RULES
 !

HEDGE "Cap" _
 TYPE CAP _
 LEG "FLT" DEAL_RECEIVES OPTIMAL_INTPMT "CAP_IN"

!
 HEDGE "Swap" _
 TYPE SWAP _
 LEG "OUT_LEG" DEAL_PAYS OPTIMAL_INTPMT "SWAP_OUT" _
 LEG "IN_LEG" DEAL_RECEIVES OPTIMAL_INTPMT "SWAP_IN"

!
 HEDGE "Cap2" _
 TYPE CAP _
 LEG "FLT" DEAL_RECEIVES OPTIMAL_INTPMT "CAP_IN2"

!

```

HEDGE "Swap2" _
    TYPE SWAP _
    LEG "OUT_LEG" DEAL_PAYS OPTIMAL_INTPMT "SWAP_OUT2" _
    LEG "IN_LEG" DEAL_RECEIVES OPTIMAL_INTPMT "SWAP_IN2" _
!
HEDGE "Cap3" _
    TYPE CAP _
    LEG "FLT" DEAL_RECEIVES OPTIMAL_INTPMT "CAP_IN3"
!
HEDGE "Swap3" _
    TYPE SWAP _
    LEG "OUT_LEG" DEAL_PAYS OPTIMAL_INTPMT "SWAP_OUT3" _
    LEG "IN_LEG" DEAL_RECEIVES OPTIMAL_INTPMT "SWAP_IN3" _
!
HEDGE "Cap4" _
    TYPE CAP _
    LEG "FLT" DEAL_RECEIVES OPTIMAL_INTPMT "CAP_IN4"
!
HEDGE "Swap4" _
    TYPE SWAP _
    LEG "OUT_LEG" DEAL_PAYS OPTIMAL_INTPMT "SWAP_OUT4" _
    LEG "IN_LEG" DEAL_RECEIVES OPTIMAL_INTPMT "SWAP_IN4" _
!

CLASS "X" NO_BUILD_TRANCHE _
    = "INV_IO"
CLASS "X1" NO_BUILD_TRANCHE _
    = "NAS_IO"
CLASS "AAA_11" NO_BUILD_TRANCHE _
    SHORTFALL_PAYBACK COUPONCAP TRUE _
    SHORTFALL_EARN_INT COUPONCAP TRUE _
    = "1A1"
CLASS "AAA_12" NO_BUILD_TRANCHE _
    SHORTFALL_PAYBACK COUPONCAP TRUE _
    SHORTFALL_EARN_INT COUPONCAP TRUE _
    = "1A2"
CLASS "AAA_2" NO_BUILD_TRANCHE _
    SHORTFALL_PAYBACK COUPONCAP TRUE _
    SHORTFALL_EARN_INT COUPONCAP TRUE _
    = "2A"
CLASS "AAA_3" NO_BUILD_TRANCHE _
    SHORTFALL_PAYBACK COUPONCAP TRUE _
    SHORTFALL_EARN_INT COUPONCAP TRUE _
    = "3A"
CLASS "AA" NO_BUILD_TRANCHE _
    SHORTFALL_PAYBACK COUPONCAP TRUE _
    SHORTFALL_EARN_INT COUPONCAP TRUE _
    = "AA"
CLASS "A" NO_BUILD_TRANCHE _
    SHORTFALL_PAYBACK COUPONCAP TRUE _
    SHORTFALL_EARN_INT COUPONCAP TRUE _
    = "A"
CLASS "A-" NO_BUILD_TRANCHE _
    SHORTFALL_PAYBACK COUPONCAP TRUE _
    SHORTFALL_EARN_INT COUPONCAP TRUE _

```

```

        = "A-"
CLASS "BBB+" NO_BUILD_TRANCHE _
    SHORTFALL_PAYBACK COUPONCAP TRUE _
    SHORTFALL_EARN_INT COUPONCAP TRUE _
    = "BBB+"
CLASS "BBB" NO_BUILD_TRANCHE _
    SHORTFALL_PAYBACK COUPONCAP TRUE _
    SHORTFALL_EARN_INT COUPONCAP TRUE _
    = "BBB"
CLASS "BBB-" NO_BUILD_TRANCHE _
    SHORTFALL_PAYBACK COUPONCAP TRUE _
    SHORTFALL_EARN_INT COUPONCAP TRUE _
    = "BBB-"
CLASS "RESID" = "R#1" "R_PP#1"
CLASS "NIM_R" NO_BUILD_TRANCHE _
    = "NIM_R"
CLASS "NIM_1" NO_BUILD_TRANCHE _
    = "NIM"
CLASS "AAA_1" ALLOCATION _
    = "AAA_11" "AAA_12"
CLASS "AAA" DISTRIB_CLASS PRORATA WRITEDOWN_BAL PRORATA ALLOCATION _
    = "AAA_1" "AAA_2" "AAA_3"
!
!
CLASS "ROOT" _
    WRITEDOWN_BAL RULES _
    DISTRIB_CLASS RULES _
    SHORTFALL_EARN_INT INTEREST TRUE _
    = "X1" "AAA" "X" "AA" "A" "A-" "BBB+" "BBB" "BBB-" "RESID" "NIM_1" "NIM_R"
!
!
    DEFINE PSEUDO_TRANCHE CLASS "AAA" Delay 0 Dated 20040210 Next 20040225
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
!
    DEFINE PSEUDO_TRANCHE CLASS "AAA_1" Delay 0 Dated 20040210 Next 20040225
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
!
!
    CROSSOVER When 0
!
!
    TRIGGER "StepDown-DlqRatio" _
        FULL_NAME "Step Down Delinquency Trigger" _
        ORIG_TESTVAL 0.000% _
        TESTVAL (#SDTrigDelinqFrac); _
        ORIG_TARGETVAL 1% _
        TARGETVAL (1%); _
        TRIGVAL LODIFF
!
    TRIGGER "STEPDOWN_TRIGGER" _
        FULL_NAME "Step Down Trigger" _
        DEFINITION "A Step Down Trigger exists, if _
;(1) the quotient of (A) the aggregate principal balance of all _
mortgage loans 90 or more days delinquent and (B) the principal _
balance of the loans, exceeds 1%." _
        IMPACT "If a Step Down Trigger is in effect the OC target CANNOT stepdown to _
5.2% of the current balance of the collateral." _

```

```

    TRIGVAL FORMULA (TRIGGER("StepDown-DlqRatio","TRIGVAL"));
!
OPTIONAL REDEMPTION: "CLEANUP" _
    COLL_FRAC 20% _
    PRICE_P ( COLL_BAL ); _
    DISTR_P RULES "OPTR_DEAL"
!
!
INTEREST_SHORTFALL FULL_PREPAY  Compensate Pro_rate _
    PARTIAL_PREPAY Compensate Pro_rate _
    LOSS          Compensate Pro_rate
!
!
CMO Block Payment Rules
-----
    from : CASH_ACCOUNT (100)
    subject to : CEILING ( ( COLL("PREPAYPENALTY") ) )
    pay : CREDIT_ENHANCEMENT ("YmRsvFnd")
-----
    calculate : #HedgePaySave = 0.00 + MAX(0.00, -1 * HEDGE("SWAP","OPTIMAL_PMT")) + MAX(0.00,
-1 * HEDGE("SWAP2","OPTIMAL_PMT")) + MAX(0.00, -1 * HEDGE("SWAP3","OPTIMAL_PMT")) +
MAX(0.00, -1 * HEDGE("SWAP4","OPTIMAL_PMT"))
-----
    from : HEDGE ("CAP")
    pay : CASH_ACCOUNT(0)
-----
    from : HEDGE ("CAP2")
    pay : CASH_ACCOUNT(0)
-----
    from : HEDGE ("CAP3")
    pay : CASH_ACCOUNT(0)
-----
    from : HEDGE ("CAP4")
    pay : CASH_ACCOUNT(0)
-----
    from : CASH_ACCOUNT (100)
    pay : HEDGE ("SWAP")
-----
    from : CASH_ACCOUNT (100)
    pay : HEDGE ("SWAP2")
-----
    from : CASH_ACCOUNT (100)
    pay : HEDGE ("SWAP3")
-----
    from : CASH_ACCOUNT (100)
    pay : HEDGE ("SWAP4")
-----
    calculate : #PrincFrac1      = COLL_P(1) / COLL_P
    calculate : #PrincFrac2      = COLL_P(2) / COLL_P
    calculate : #PrincFrac3      = COLL_P(3) / COLL_P
!
    calculate : #XtraPFrac1      = COLL_P(1) / COLL_P
    calculate : #XtraPFrac2      = COLL_P(2) / COLL_P
    calculate : #XtraPFrac3      = COLL_P(3) / COLL_P
!

```

```

calculate : #Princ          = COLL_P
!
calculate : #Interest      = COLL_I + HEDGE("CAP","ACTUAL_PMT") +
HEDGE("CAP2","ACTUAL_PMT") + HEDGE("CAP3","ACTUAL_PMT") +
HEDGE("CAP4","ACTUAL_PMT")
!
calculate : #PrevSpecOC    = #SpecOCTarg
!
calculate : #CurrentOC     = MAX( 0, COLL_BAL - (BBAL("1A1#1", "1A2#1", "2A#1", "3A#1",
"AA#1", "A#1", "A-#1", "BBB+#1", "BBB#1", "BBB-#1") - #Princ))
!
calculate : #XSSpread      = MAX( 0, #Interest - ( COLL_YM * (100% - 0%)) -
(OPTIMAL_INTPMT("ROOT") - OPTIMAL_INTPMT("NIM")) - (INTSHORT_ACCUM("ROOT") -
INTSHORT_ACCUM("NIM")) + (COUPONCAP_SHORTFALL("ROOT") -
COUPONCAP_SHORTFALL("NIM"))) )
!
calculate : #FloorOCTotal  = #FloorOCTarg
!
calculate : #StepOCTarg    = COLL_BAL * #StepOCFrac
!
calculate : #StepDownDatePass = CURMONTH GE #StepDownDate
!
calculate : #StepDownBal    = COLL_BAL < #OrigCollBal * 50%
!
calculate : #SDTrigDelinqFrac = AVG_COLL("RATE",-1,3,1)
!
calculate : #SDTrigEvent    = TRIGGER("STEPDOWN_TRIGGER")
!
calculate : #StepDown       = #StepDown OR (( #StepDownDatePass AND #StepDownBal ) AND
NOT #SDTrigEvent )
!
calculate : #SpecOCTarg     = IF #StepDown _
                           THEN 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 - ( COLL_YM * (100% - 0%)) -
(OPTIMAL_INTPMT("ROOT") - OPTIMAL_INTPMT("NIM")) - (INTSHORT_ACCUM("ROOT") -
INTSHORT_ACCUM("NIM")) + #OCSurplus + (COUPONCAP_SHORTFALL("ROOT") -
COUPONCAP_SHORTFALL("NIM"))) )
!
calculate : #SubDefic       = MAX ( 0, ( (BBAL("ROOT") - BBAL("NIM")) - #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 : #ClassAAPDA      = IF ( #StepDown EQ 0.0 ) _
                             THEN #DistribAmt _
                             ELSE BBAL("1A1", "1A2", "2A", "3A") _
                             - MIN(COLL_BAL - #FloorOCTotal, #AAATargPct * COLL_BAL)
calculate : #ClassAAPDA      = MAX( 0.0, MIN(BBAL("1A1", "1A2", "2A", "3A"), #ClassAAPDA ) )
calculate : #ClassAAPDA      = MAX( 0, MIN( #ClassAAPDA, #DistribAmt ) )
!
!
calculate : #ClassAAPDA      = IF ( #StepDown EQ 0.0 ) _
                             THEN #DistribAmt - #ClassAAPDA _
                             ELSE BBAL("1A1", "1A2", "2A", "3A", "AA") - #ClassAAPDA _
                             - MIN(COLL_BAL - #FloorOCTotal, #AATargPct * COLL_BAL)
calculate : #ClassAAPDA      = MAX( 0.0, MIN(BBAL("AA"), #ClassAAPDA ) )
calculate : #ClassAAPDA      = MAX( 0, MIN( #ClassAAPDA, #DistribAmt - #ClassAAPDA ) )
!
!
calculate : #ClassAPDA       = IF ( #StepDown EQ 0.0 ) _
                             THEN #DistribAmt - #ClassAAPDA - #ClassAAPDA _
                             ELSE BBAL("1A1", "1A2", "2A", "3A", "AA", "A") - #ClassAAPDA -
#ClassAAPDA _
                             - MIN(COLL_BAL - #FloorOCTotal, #ATargPct * COLL_BAL)
calculate : #ClassAPDA       = MAX( 0.0, MIN(BBAL("A"), #ClassAPDA ) )
calculate : #ClassAPDA       = MAX( 0, MIN( #ClassAPDA, #DistribAmt - #ClassAAPDA -
#ClassAAPDA ) )
!
!
calculate : #ClassAMPDA      = IF ( #StepDown EQ 0.0 ) _
                             THEN #DistribAmt - #ClassAAPDA - #ClassAAPDA - #ClassAPDA _
                             ELSE BBAL("1A1", "1A2", "2A", "3A", "AA", "A", "A-") - #ClassAAPDA -
#ClassAAPDA - #ClassAPDA _
                             - MIN(COLL_BAL - #FloorOCTotal, #AMTargPct * COLL_BAL)
calculate : #ClassAMPDA      = MAX( 0.0, MIN(BBAL("A-"), #ClassAMPDA ) )
calculate : #ClassAMPDA      = MAX( 0, MIN( #ClassAMPDA, #DistribAmt - #ClassAAPDA -
#ClassAAPDA - #ClassAPDA ) )
!
!
calculate : #ClassBBBPPDA    = IF ( #StepDown EQ 0.0 ) _
                             THEN #DistribAmt - #ClassAAPDA - #ClassAAPDA - #ClassAPDA -
#ClassAMPDA _
                             ELSE BBAL("1A1", "1A2", "2A", "3A", "AA", "A", "A-", "BBB+") - #ClassAAPDA
- #ClassAAPDA - #ClassAPDA - #ClassAMPDA _
                             - MIN(COLL_BAL - #FloorOCTotal, #BBBPTargPct * COLL_BAL)
calculate : #ClassBBBPPDA    = MAX( 0.0, MIN(BBAL("BBB+"), #ClassBBBPPDA ) )
calculate : #ClassBBBPPDA    = MAX( 0, MIN( #ClassBBBPPDA, #DistribAmt - #ClassAAPDA -
#ClassAAPDA - #ClassAPDA - #ClassAMPDA ) )
!
!
calculate : #ClassBBBPPDA    = IF ( #StepDown EQ 0.0 ) _
                             THEN #DistribAmt - #ClassAAPDA - #ClassAAPDA - #ClassAPDA -

```

```

#ClassAMPDA - #ClassBBBPPDA _
      ELSE BBAL("1A1", "1A2", "2A", "3A", "AA", "A", "A-", "BBB+", "BBB") -
#ClassAAPDA - #ClassAAPDA - #ClassAPDA - #ClassAMPDA - #ClassBBBPPDA _
      - MIN(COLL_BAL - #FloorOCTotal, #BBBTargPct * COLL_BAL)
      calculate : #ClassBBBPDA      = MAX( 0.0, MIN(BBAL("BBB"), #ClassBBBPDA ))
      calculate : #ClassBBBPDA      = MAX( 0, MIN( #ClassBBBPDA, #DistribAmt - #ClassAAPDA -
#ClassAAPDA - #ClassAPDA - #ClassAMPDA - #ClassBBBPPDA ) )
!
!
      calculate : #ClassBBBMPDA      = IF (#StepDown EQ 0.0) _
      THEN #DistribAmt - #ClassAAPDA - #ClassAAPDA - #ClassAPDA -
#ClassAMPDA - #ClassBBBPPDA - #ClassBBBPDA _
      ELSE BBAL("1A1", "1A2", "2A", "3A", "AA", "A", "A-", "BBB+", "BBB", "BBB-") -
#ClassAAPDA - #ClassAAPDA - #ClassAPDA - #ClassAMPDA - #ClassBBBPPDA - #ClassBBBPDA _
      - MIN(COLL_BAL - #FloorOCTotal, #BBBMTargPct * COLL_BAL)
      calculate : #ClassBBBMPDA      = MAX( 0.0, MIN(BBAL("BBB-"), #ClassBBBMPDA ))
      calculate : #ClassBBBMPDA      = MAX( 0, MIN( #ClassBBBMPDA, #DistribAmt - #ClassAAPDA -
#ClassAAPDA - #ClassAPDA - #ClassAMPDA - #ClassBBBPPDA - #ClassBBBPDA ) )
!
!
      calculate : "AAA" _
NO_CHECK CUSTOM AMOUNT      = #ClassAAPDA
!
      calculate : "AA" _
NO_CHECK CUSTOM AMOUNT      = #ClassAAPDA
!
      calculate : "A" _
NO_CHECK CUSTOM AMOUNT      = #ClassAPDA
!
      calculate : "A-" _
NO_CHECK CUSTOM AMOUNT      = #ClassAMPDA
!
      calculate : "BBB+" _
NO_CHECK CUSTOM AMOUNT      = #ClassBBBPPDA
!
      calculate : "BBB" _
NO_CHECK CUSTOM AMOUNT      = #ClassBBBPDA
!
      calculate : "BBB-" _
NO_CHECK CUSTOM AMOUNT      = #ClassBBBMPDA
!
-----
      pay : CLASS INTEREST SEQUENTIAL ("X")
      pay : CLASS INTSHORT SEQUENTIAL ("X")
-----
      pay : CLASS INTEREST SEQUENTIAL ("X1")
      pay : CLASS INTSHORT SEQUENTIAL ("X1")
-----
      pay : CLASS INTEREST PRO_RATE ("AAA")
      pay : CLASS INTSHORT PRO_RATE ("AAA")
      pay : CLASS PRINCIPAL SEQUENTIAL ("AAA")
      pay : CLASS INTEREST PRO_RATE ("AA")
      pay : CLASS INTSHORT PRO_RATE ("AA")
      pay : CLASS PRINCIPAL SEQUENTIAL ("AA")
      pay : CLASS INTEREST PRO_RATE ("A")

```

```

pay : CLASS INTSHORT PRO_RATE ( "A" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "A" )
pay : CLASS INTEREST PRO_RATE ( "A-" )
pay : CLASS INTSHORT PRO_RATE ( "A-" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "A-" )
pay : CLASS INTEREST PRO_RATE ( "BBB+" )
pay : CLASS INTSHORT PRO_RATE ( "BBB+" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "BBB+" )
pay : CLASS INTEREST PRO_RATE ( "BBB" )
pay : CLASS INTSHORT PRO_RATE ( "BBB" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "BBB" )
pay : CLASS INTEREST PRO_RATE ( "BBB-" )
pay : CLASS INTSHORT PRO_RATE ( "BBB-" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "BBB-" )

-----
calculate : #SeniorPrinc      = #PrincPmt / #DistribAmt * #ClassAAAPDA
calculate : #SeniorXtraP     = #ClassAAAPDA - #SeniorPrinc
!
calculate : #SeniorPDA1      = MIN( BBAL("AAA_1"), (#SeniorPrinc * #PrincFrac1) + (#SeniorXtraP *
#XtraPFrac1))
calculate : #SeniorPDA2      = MIN( BBAL("AAA_2"), (#SeniorPrinc * #PrincFrac2) + (#SeniorXtraP *
#XtraPFrac2))
calculate : #SeniorPDA3      = MIN( BBAL("AAA_3"), (#SeniorPrinc * #PrincFrac3) + (#SeniorXtraP *
#XtraPFrac3))

-----
from : CLASS ( "AAA" )
pay : CLASS INTEREST PRO_RATE ( "AAA_1"; "AAA_2"; "AAA_3" )
pay : CLASS INTSHORT PRO_RATE ( "AAA_1"; "AAA_2"; "AAA_3" )

-----
from : SUBACCOUNT ( #SeniorPDA1, CLASS "AAA" )
pay : CLASS BALANCE SEQUENTIAL ( "AAA_1" )

-----
from : SUBACCOUNT ( #SeniorPDA2, CLASS "AAA" )
pay : CLASS BALANCE SEQUENTIAL ( "AAA_2" )

-----
from : SUBACCOUNT ( #SeniorPDA3, CLASS "AAA" )
pay : CLASS BALANCE SEQUENTIAL ( "AAA_3" )

-----
from : CLASS ( "AAA" )
pay : CLASS BALANCE PRO_RATE ( "AAA_1"; "AAA_2"; "AAA_3" )

-----
!
calculate : #P_AAA_1 = BBAL("AAA_11","AAA_12") - BBAL("AAA_1")

-----
from : CLASS ( "AAA_1" )
pay : CLASS INTEREST SEQUENTIAL ( "AAA_11" )
pay : CLASS INTSHORT SEQUENTIAL ( "AAA_11" )

-----
from : CLASS ( "AAA_1" )
from : SUBACCOUNT ( #P_AAA_1 )
pay : CLASS BALANCE SEQUENTIAL ( "AAA_11" )

-----
from : CLASS ( "AAA_1" )
pay : CLASS INTEREST SEQUENTIAL ( "AAA_12" )
pay : CLASS INTSHORT SEQUENTIAL ( "AAA_12" )

```

```

-----
from : CLASS ( "AAA_1" )
from : SUBACCOUNT ( #P_AAA_1 )
pay : CLASS BALANCE SEQUENTIAL ( "AAA_12" )
-----
!
-----
from : CLASS ( "AAA_11" )
pay : SEQUENTIAL ( "1A1#1" )
-----
from : CLASS ( "AAA_12" )
pay : SEQUENTIAL ( "1A2#1" )
-----
from : CLASS ( "AAA_2" )
pay : SEQUENTIAL ( "2A#1" )
-----
from : CLASS ( "AAA_3" )
pay : SEQUENTIAL ( "3A#1" )
-----
from : CLASS ( "AA" )
pay : SEQUENTIAL ( "AA#1" )
-----
from : CLASS ( "A" )
pay : SEQUENTIAL ( "A#1" )
-----
from : CLASS ( "A-" )
pay : SEQUENTIAL ( "A-#1" )
-----
from : CLASS ( "BBB+" )
pay : SEQUENTIAL ( "BBB+#1" )
-----
from : CLASS ( "BBB" )
pay : SEQUENTIAL ( "BBB#1" )
-----
from : CLASS ( "BBB-" )
pay : SEQUENTIAL ( "BBB-#1" )
-----
!
!! Penalty Allocation
!
-----
from : CREDIT_ENHANCEMENT ( "YmRsvFnd" )
subject to : CEILING ( ( COLL("PREPAYPENALTY") * 100% ) )
pay : PREPAYPENALTY SEQUENTIAL ( "R_PP#1" )
-----
!
from : CLASS ( "ROOT" )
pay : CLASS COUPONCAP_SHORT PRO_RATA ( "AAA_11" ; "AAA_12" ; "AAA_2" ; "AAA_3" ;
"AA" ; "A" ; "A-" ; "BBB+" ; "BBB" ; "BBB-" )

```

```

-----
from : HEDGE ("SWAP")
pay : CLASS COUPONCAP_SHORT PRO_RATA("AAA_11" ; "AAA_12" ; "AAA_2" ; "AAA_3")
-----
from : HEDGE ("SWAP")
pay : CLASS COUPONCAP_SHORT PRO_RATA("AA")
-----
from : HEDGE ("SWAP")
pay : CLASS COUPONCAP_SHORT PRO_RATA("A")
-----
from : HEDGE ("SWAP")
pay : CLASS COUPONCAP_SHORT PRO_RATA("A-")
-----
from : HEDGE ("SWAP")
pay : CLASS COUPONCAP_SHORT PRO_RATA("BBB+")
-----
from : HEDGE ("SWAP")
pay : CLASS COUPONCAP_SHORT PRO_RATA("BBB")
-----
from : HEDGE ("SWAP")
pay : CLASS COUPONCAP_SHORT PRO_RATA("BBB-")
-----
from : HEDGE ("SWAP2")
pay : CLASS COUPONCAP_SHORT PRO_RATA("AAA_11" ; "AAA_12" ; "AAA_2" ; "AAA_3")
-----
from : HEDGE ("SWAP2")
pay : CLASS COUPONCAP_SHORT PRO_RATA("AA")
-----
from : HEDGE ("SWAP2")
pay : CLASS COUPONCAP_SHORT PRO_RATA("A")
-----
from : HEDGE ("SWAP2")
pay : CLASS COUPONCAP_SHORT PRO_RATA("A-")
-----
from : HEDGE ("SWAP2")
pay : CLASS COUPONCAP_SHORT PRO_RATA("BBB+")
-----
from : HEDGE ("SWAP2")
pay : CLASS COUPONCAP_SHORT PRO_RATA("BBB")
-----
from : HEDGE ("SWAP2")
pay : CLASS COUPONCAP_SHORT PRO_RATA("BBB-")
-----
from : HEDGE ("SWAP3")
pay : CLASS COUPONCAP_SHORT PRO_RATA("AAA_11" ; "AAA_12" ; "AAA_2" ; "AAA_3")
-----
from : HEDGE ("SWAP3")
pay : CLASS COUPONCAP_SHORT PRO_RATA("AA")
-----
from : HEDGE ("SWAP3")
pay : CLASS COUPONCAP_SHORT PRO_RATA("A")
-----
from : HEDGE ("SWAP3")
pay : CLASS COUPONCAP_SHORT PRO_RATA("A-")
-----

```

```

from : HEDGE ("SWAP3")
pay : CLASS COUPONCAP_SHORT PRO_RATE("BBB+")
-----
from : HEDGE ("SWAP3")
pay : CLASS COUPONCAP_SHORT PRO_RATE("BBB")
-----
from : HEDGE ("SWAP3")
pay : CLASS COUPONCAP_SHORT PRO_RATE("BBB-")
-----
from : HEDGE ("SWAP4")
pay : CLASS COUPONCAP_SHORT PRO_RATE("AAA_11"; "AAA_12"; "AAA_2"; "AAA_3")
-----
from : HEDGE ("SWAP4")
pay : CLASS COUPONCAP_SHORT PRO_RATE("AA")
-----
from : HEDGE ("SWAP4")
pay : CLASS COUPONCAP_SHORT PRO_RATE("A")
-----
from : HEDGE ("SWAP4")
pay : CLASS COUPONCAP_SHORT PRO_RATE("A-")
-----
from : HEDGE ("SWAP4")
pay : CLASS COUPONCAP_SHORT PRO_RATE("BBB+")
-----
from : HEDGE ("SWAP4")
pay : CLASS COUPONCAP_SHORT PRO_RATE("BBB")
-----
from : HEDGE ("SWAP4")
pay : CLASS COUPONCAP_SHORT PRO_RATE("BBB-")
-----
from : HEDGE ("SWAP")
pay : AS_INTEREST ("R#1")
-----
from : HEDGE ("SWAP2")
pay : AS_INTEREST ("R#1")
-----
from : HEDGE ("SWAP3")
pay : AS_INTEREST ("R#1")
-----
from : HEDGE ("SWAP4")
pay : AS_INTEREST ("R#1")
-----
!
from : CLASS ("ROOT")
pay : AS_INTEREST ("R#1")
-----
calculate : #WriteDown = MAX(0.0,
BBAL("1A1#1","1A2#1","2A#1","3A#1","AA#1","A#1","A-#1","BBB+#1","BBB#1","BBB-#1") - COLL_BAL)
-----
from : SUBACCOUNT ( #Writedown )
pay : WRITEDOWN SEQUENTIAL ( "BBB-#1" )
-----
from : SUBACCOUNT ( #Writedown )
pay : WRITEDOWN SEQUENTIAL ( "BBB#1" )
-----

```

```

from : SUBACCOUNT ( #Writedown )
pay : WRITEDOWN SEQUENTIAL ( "BBB+#1" )
-----
from : SUBACCOUNT ( #Writedown )
pay : WRITEDOWN SEQUENTIAL ( "A-#1" )
-----
from : SUBACCOUNT ( #Writedown )
pay : WRITEDOWN SEQUENTIAL ( "A#1" )
-----
from : SUBACCOUNT ( #Writedown )
pay : WRITEDOWN SEQUENTIAL ( "AA#1" )
-----
calculate : #BondBal =
BBAL("1A1#1","1A2#1","2A#1","3A#1","AA#1","A#1","A-#1","BBB+#1","BBB#1","BBB-#1")
calculate : #OC = MAX( 0, COLL_BAL - #BondBal )
-----
calculate : #Cash_NIM_1 = IF IS_PAYMONTH("NIM#1") THEN INTPMT("R") + INTPMT("R_PP")
ELSE 0
-----
from : SUBACCOUNT ( #Cash_NIM_1, NOWHERE NOLIMIT )
pay : CLASS INTEREST SEQUENTIAL ( "NIM_1" )
pay : CLASS BALANCE SEQUENTIAL ( "NIM_1" )
-----
pay : SEQUENTIAL ( "NIM#1" )
-----
from : SUBACCOUNT ( #Cash_NIM_1, NOWHERE NOLIMIT )
pay : AS_INTEREST ( "NIM_R#1" )
-----
calculate : #IntShort_NIM = MAX( 0, OPTIMAL_INTPMT( "NIM" ) - INTPMT ( "NIM" ))
-----
pay : INCREMENT ( BALANCE "NIM" , BY #IntShort_NIM )
-----
calculate: #CallBalDeal = COLL_BAL
-----
----- SECTION: "OPTR_DEAL"
-----
from : CASH_ACCOUNT (100)
subject to : CEILING ( #CallBalDeal)
pay : CLASS BALANCE SEQUENTIAL ( "ROOT" )
pay : CLASS MORE_INTEREST SEQUENTIAL ( "ROOT" )
-----
from : CLASS ( "ROOT" )
pay : SEQUENTIAL ( "1A1#1", "1A2#1", "2A#1", "3A#1", "AA#1", "A#1", "A-#1", "BBB+#1",
"BBB#1", "BBB-#1" )
-----
from : CLASS ( "ROOT" )
pay : AS_INTEREST ( "R#1" )
-----
calculate : #Cash_NIM_1 = IF IS_PAYMONTH("NIM#1") THEN INTPMT("R") + INTPMT("R_PP")
ELSE 0
-----
from : SUBACCOUNT ( #Cash_NIM_1, NOWHERE NOLIMIT )
pay : CLASS INTEREST SEQUENTIAL ( "NIM_1" )
pay : CLASS BALANCE SEQUENTIAL ( "NIM_1" )

```

```

-----
pay : SEQUENTIAL ( "NIM#1" )
-----
from : SUBACCOUNT ( #Cash_NIM_1, NOWHERE NOLIMIT )
pay : AS_INTEREST ( "NIM_R#1" )
-----
calculate : #IntShort_NIM = MAX( 0, OPTIMAL_INTPMT( "NIM" ) - INTPMT ( "NIM" ))
-----
pay : INCREMENT ( BALANCE "NIM" , BY #IntShort_NIM )
-----

```

Schedule "Fieldstone04-1CM4b"

DECLARE

VALUES OK

20040225	187,235,889.77
20040325	173,419,799.37
20040425	160,622,892.10
20040525	148,770,005.48
20040625	137,791,518.76
20040725	127,622,944.32
20040825	118,204,549.38
20040925	109,481,005.45
20041025	101,401,063.66
20041125	93,917,254.05
20041225	86,985,606.97
20050125	80,565,395.05
20050225	74,618,894.15
20050325	69,111,162.00
20050425	64,009,833.06
20050525	59,284,928.61
20050625	54,908,680.69
20050725	50,855,369.21
20050825	0

!

Schedule "NAS_IO_BAL"

DECLARE

VALUES OK

20040225	173,419,799.37
20040325	160,622,892.10
20040425	148,770,005.48
20040525	137,791,518.76
20040625	127,622,944.32
20040725	118,204,549.38
20040825	109,481,005.45
20040925	101,401,063.66
20041025	93,917,254.05
20041125	86,985,606.97
20041225	80,565,395.05
20050125	74,618,894.15
20050225	69,111,162.00
20050325	64,009,833.06
20050425	59,284,928.61
20050525	54,908,680.69
20050625	50,855,369.21
20050725	0

!

```

!
!
! DEFINE DYNAMIC #YM_6MONTH95100 = LOAN("SCHAM_PREP_AMT") * 95% *
! LOAN("GROSSRATE") / 1200 * 6
!
! Collateral OVER
!
! Factor --Delay--
! Type Date P/Y BV Use BV for 0
! WL 20040101 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
! ToRst RstPer Cap Cap Floor Negam Back
!! BEGINNING OF COLLATERAL
M 1 "Arm2280" WL 00 WAC 7.976 ( 15205262.21 / 15205262.21 );
15205262.21 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.800 24 6 SYNC_INT 13.976 1.000 7.976 0 0 INIT_PERCAP 3.000
GROUP "2A" PREPAY_FLAG NONE ; TEASER
M 2 "Arm2280" WL 00 WAC 7.517 ( 2885990.89 / 2885990.89 );
2885990.89 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.732 24 6 SYNC_INT 13.517 1.000 7.517 0 0 INIT_PERCAP 3.000
GROUP "2A" PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA ( #YM_6MONTH95100 ); TEASER
M 3 "Arm2281" WL 00 WAC 7.287 ( 1259724.91 / 1259724.91 );
1259724.91 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
6.051 24 6 SYNC_INT 13.287 1.000 7.287 0 0 INIT_PERCAP 3.000
GROUP "2A" PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA ( #YM_6MONTH95100 ); TEASER
M 4 "Arm2282" WL 00 WAC 7.231 ( 90351411.79 / 90351411.79 );
90351411.79 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.690 24 6 SYNC_INT 13.231 1.000 7.231 0 0 INIT_PERCAP 3.000
GROUP "2A" PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA ( #YM_6MONTH95100 ); TEASER
M 5 "Arm2282" WL 00 WAC 7.699 ( 2312634.44 / 2312634.44 );
2312634.44 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
6.026 24 6 SYNC_INT 13.699 1.000 7.699 0 0 INIT_PERCAP 3.000
GROUP "2A" PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA ( #YM_6MONTH95100 ); TEASER
M 6 "Arm2283" WL 00 WAC 7.529 ( 646463.01 / 646463.01 ); 646463.01
.50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 5.561 24 6
SYNC_INT 13.529 1.000 7.529 0 0 INIT_PERCAP 3.000
GROUP "2A" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_6MONTH95100 ); TEASER
M 7 "Arm3270" WL 00 WAC 7.982 ( 640930.50 / 640930.50 ); 640930.50
.50 .50 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO 5.773 35 6
SYNC_INT 13.982 1.000 7.982 0 0 INIT_PERCAP 3.000
GROUP "2A" PREPAY_FLAG NONE ; TEASER
M 8 "Arm3273" WL 00 WAC 6.891 ( 2465675.26 / 2465675.26 );
2465675.26 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.631 36 6 SYNC_INT 12.891 1.000 6.891 0 0 INIT_PERCAP 3.000
GROUP "2A" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_6MONTH95100 ); TEASER
M 9 "Arm5252" WL 00 WAC 5.500 ( 208000.00 / 208000.00 ); 208000.00
.50 .50 360:0 360:0 360 NO_CHECK ARM CMT_1YR 5.500 61 12
SYNC_INT 11.500 2.000 5.500 0 0 GROUP
"2A" PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA ( #YM_6MONTH95100 ); TEASER
M 10 "Arm5253" WL 00 WAC 6.363 ( 1224416.66 / 1224416.66 );
1224416.66 .50 .50 359:1 359:1 360 NO_CHECK ARM CMT_1YR
5.650 60 12 SYNC_INT 12.363 2.000 6.363 0 0

```

GROUP "2A" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 11 "Arm2280" WL 00 WAC 7.661 (5212323.00 / 5212323.00);
5212323.00 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.735 24 6 SYNC_INT 13.661 1.000 7.661 0 0 INIT_PERCAP 3.000
AMORT NONE FOR 60 GROUP "2B" PREPAY_FLAG NONE ; TEASER
M 12 "Arm2280" WL 00 WAC 6.698 (357750.00 / 357750.00); 357750.00
.50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 5.692 24 6
SYNC_INT 12.698 1.000 6.698 0 0 INIT_PERCAP 3.000 AMORT NONE
FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (#YM_6MONTH95100
); TEASER
M 13 "Arm2281" WL 00 WAC 7.182 (1341970.00 / 1341970.00);
1341970.00 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
6.034 24 6 SYNC_INT 13.182 1.000 7.182 0 0 INIT_PERCAP 3.000
AMORT NONE FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 14 "Arm2282" WL 00 WAC 6.639 (135958295.12 / 135958295.12);
135958295.12 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.550 24 6 SYNC_INT 12.639 1.000 6.639 0 0 INIT_PERCAP 3.000
AMORT NONE FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 15 "Arm2282" WL 00 WAC 7.319 (756900.00 / 756900.00); 756900.00
.50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 6.038 24 6
SYNC_INT 13.319 1.000 7.319 0 0 INIT_PERCAP 3.000 AMORT NONE
FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 16 "Arm2283" WL 00 WAC 6.399 (1104950.00 / 1104950.00);
1104950.00 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.567 24 6 SYNC_INT 12.399 1.000 6.399 0 0 INIT_PERCAP 3.000
AMORT NONE FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 17 "Arm3270" WL 00 WAC 7.875 (175200.00 / 175200.00); 175200.00
.50 .50 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO 5.200 37 6
SYNC_INT 13.875 1.000 7.875 0 0 INIT_PERCAP 3.000 AMORT NONE
FOR 60 GROUP "2B" PREPAY_FLAG NONE ; TEASER
M 18 "Arm3273" WL 00 WAC 6.765 (6000847.83 / 6000847.83);
6000847.83 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.609 36 6 SYNC_INT 12.765 1.000 6.765 0 0 INIT_PERCAP 3.000
AMORT NONE FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 19 "Arm5252" WL 00 WAC 6.300 (279000.00 / 279000.00); 279000.00
.50 .50 360:0 360:0 360 NO_CHECK ARM CMT_1YR 6.000 61 12
SYNC_INT 12.300 2.000 6.300 0 0 AMORT NONE FOR
60 GROUP "2B" PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_6MONTH95100);
TEASER
M 20 "Arm5253" WL 00 WAC 6.467 (1780120.00 / 1780120.00);
1780120.00 .50 .50 359:1 359:1 360 NO_CHECK ARM CMT_1YR
5.500 60 12 SYNC_INT 12.467 2.000 6.467 0 0 AMORT
NONE FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 21 "Arm2280" WL 00 WAC 8.008 (14266721.15 / 14266721.15);
14266721.15 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.894 24 6 SYNC_INT 14.008 1.000 8.008 0 0 INIT_PERCAP 3.000
GROUP "2A" PREPAY_FLAG NONE ; TEASER
M 22 "Arm2280" WL 00 WAC 7.702 (2303000.15 / 2303000.15);
2303000.15 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO

5.821 24 6 SYNC_INT 13.702 1.000 7.702 0 0 INIT_PERCAP 3.000
 GROUP "2A" PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 23 "Arm2281" WL 00 WAC 8.050 (123600.00 / 123600.00); 123600.00
 .50 .50 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO 6.250 25 6
 SYNC_INT 14.050 1.000 8.050 0 0 INIT_PERCAP 3.000
 GROUP "2A" PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 24 "Arm2282" WL 00 WAC 7.208 (93071862.37 / 93071862.37);
 93071862.37 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
 5.733 24 6 SYNC_INT 13.208 1.000 7.208 0 0 INIT_PERCAP 3.000
 GROUP "2A" PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 25 "Arm2282" WL 00 WAC 7.668 (3468187.77 / 3468187.77);
 3468187.77 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
 5.984 24 6 SYNC_INT 13.668 1.000 7.668 0 0 INIT_PERCAP 3.000
 GROUP "2A" PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 26 "Arm2283" WL 00 WAC 7.047 (849070.31 / 849070.31); 849070.31
 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 5.626 24 6
 SYNC_INT 13.047 1.000 7.047 0 0 INIT_PERCAP 3.000
 GROUP "2A" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 27 "Arm3270" WL 00 WAC 8.421 (194790.94 / 194790.94); 194790.94
 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 5.883 36 6
 SYNC_INT 14.421 1.000 8.421 0 0 INIT_PERCAP 3.000
 GROUP "2A" PREPAY_FLAG NONE ; TEASER
 M 28 "Arm3270" WL 00 WAC 7.950 (85500.00 / 85500.00); 85500.00
 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 5.200 36 6
 SYNC_INT 13.950 1.000 7.950 0 0 INIT_PERCAP 3.000
 GROUP "2A" PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 29 "Arm3271" WL 00 WAC 7.500 (315000.00 / 315000.00); 315000.00
 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 6.050 36 6
 SYNC_INT 13.500 1.000 7.500 0 0 INIT_PERCAP 3.000
 GROUP "2A" PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 30 "Arm3273" WL 00 WAC 7.443 (2337592.51 / 2337592.51);
 2337592.51 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
 5.688 36 6 SYNC_INT 13.443 1.000 7.443 0 0 INIT_PERCAP 3.000
 GROUP "2A" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 31 "Arm5253" WL 00 WAC 6.836 (1710419.71 / 1710419.71);
 1710419.71 .50 .50 359:1 359:1 360 NO_CHECK ARM CMT_1YR
 5.869 60 12 SYNC_INT 12.836 2.000 6.836 0 0
 GROUP "2A" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 32 "Arm2280" WL 00 WAC 7.549 (5375860.53 / 5375860.53);
 5375860.53 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
 5.702 24 6 SYNC_INT 13.549 1.000 7.549 0 0 INIT_PERCAP 3.000
 AMORT NONE FOR 60 GROUP "2B" PREPAY_FLAG NONE ; TEASER
 M 33 "Arm2280" WL 00 WAC 7.100 (277250.00 / 277250.00); 277250.00
 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 5.800 24 6
 SYNC_INT 13.100 1.000 7.100 0 0 INIT_PERCAP 3.000 AMORT NONE
 FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (#YM_6MONTH95100
); TEASER
 M 34 "Arm2281" WL 00 WAC 7.360 (684565.00 / 684565.00); 684565.00
 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 6.153 24 6
 SYNC_INT 13.360 1.000 7.360 0 0 INIT_PERCAP 3.000 AMORT NONE
 FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 35 "Arm2282" WL 00 WAC 6.622 (133958857.42 / 133958857.42);
 133958857.42 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
 5.563 24 6 SYNC_INT 12.622 1.000 6.622 0 0 INIT_PERCAP 3.000

AMORT NONE FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 36 "Arm2282" WL 00 WAC 7.379 (1877869.00 / 1877869.00);
 1877869.00 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
 5.945 24 6 SYNC_INT 13.379 1.000 7.379 0 0 INIT_PERCAP 3.000
 AMORT NONE FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 37 "Arm2283" WL 00 WAC 5.950 (150328.00 / 150328.00); 150328.00
 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 5.500 24 6
 SYNC_INT 11.950 1.000 5.950 0 0 INIT_PERCAP 3.000 AMORT NONE
 FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 38 "Arm3270" WL 00 WAC 7.600 (135000.00 / 135000.00); 135000.00
 .50 .50 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO 6.250 35 6
 SYNC_INT 13.600 1.000 7.600 0 0 INIT_PERCAP 3.000 AMORT NONE
 FOR 60 GROUP "2B" PREPAY_FLAG NONE ; TEASER
 M 39 "Arm3272" WL 00 WAC 6.802 (492700.00 / 492700.00); 492700.00
 .50 .50 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO 5.409 35 6
 SYNC_INT 12.802 1.000 6.802 0 0 INIT_PERCAP 3.000 AMORT NONE
 FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 40 "Arm3273" WL 00 WAC 6.517 (7206008.00 / 7206008.00);
 7206008.00 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
 5.614 36 6 SYNC_INT 12.517 1.000 6.517 0 0 INIT_PERCAP 3.000
 AMORT NONE FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 41 "Arm5253" WL 00 WAC 6.565 (1283920.00 / 1283920.00);
 1283920.00 .50 .50 359:1 359:1 360 NO_CHECK ARM CMT_1YR
 5.748 60 12 SYNC_INT 12.565 2.000 6.565 0 0 AMORT
 NONE FOR 60 GROUP "2B" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 42 "Arm2280" WL 00 WAC 7.404 (2798381.03 / 2798381.03);
 2798381.03 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
 5.713 24 6 SYNC_INT 13.404 1.000 7.404 0 0 INIT_PERCAP 3.000
 GROUP "3A" PREPAY_FLAG NONE ; TEASER
 M 43 "Arm2281" WL 00 WAC 6.343 (762591.00 / 762591.00); 762591.00
 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 5.715 24 6
 SYNC_INT 12.343 1.000 6.343 0 0 INIT_PERCAP 3.000
 GROUP "3A" PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 44 "Arm2282" WL 00 WAC 6.802 (16376427.48 / 16376427.48);
 16376427.48 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
 5.551 24 6 SYNC_INT 12.802 1.000 6.802 0 0 INIT_PERCAP 3.000
 GROUP "3A" PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 45 "Arm2282" WL 00 WAC 7.590 (399500.00 / 399500.00); 399500.00
 .50 .50 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO 5.500 25 6
 SYNC_INT 13.590 1.000 7.590 0 0 INIT_PERCAP 3.000
 GROUP "3A" PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 46 "Arm3271" WL 00 WAC 7.750 (446500.00 / 446500.00); 446500.00
 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 5.500 36 6
 SYNC_INT 13.750 1.000 7.750 0 0 INIT_PERCAP 3.000
 GROUP "3A" PREPAY_FLAG YM FOR 18 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
 M 47 "Arm3273" WL 00 WAC 6.641 (1082201.17 / 1082201.17);
 1082201.17 .50 .50 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
 5.407 35 6 SYNC_INT 12.641 1.000 6.641 0 0 INIT_PERCAP 3.000
 GROUP "3A" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER

M 48 "Arm5252" WL 00 WAC 6.750 (370000.00 / 370000.00); 370000.00
.50 .50 360:0 360:0 360 NO_CHECK ARM CMT_1YR 5.850 61 12
SYNC_INT 12.750 2.000 6.750 0 0 GROUP
"3A" PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 49 "Arm5253" WL 00 WAC 6.700 (454406.23 / 454406.23); 454406.23
.50 .50 358:2 358:2 360 NO_CHECK ARM CMT_1YR 5.500 59 12
SYNC_INT 12.700 2.000 6.700 0 0 GROUP
"3A" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 50 "Arm2280" WL 00 WAC 7.452 (4980512.74 / 4980512.74);
4980512.74 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.805 24 6 SYNC_INT 13.452 1.000 7.452 0 0 INIT_PERCAP 3.000
AMORT NONE FOR 60 GROUP "3B" PREPAY_FLAG NONE ; TEASER
M 51 "Arm2281" WL 00 WAC 6.833 (730050.00 / 730050.00); 730050.00
.50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 6.227 24 6
SYNC_INT 12.833 1.000 6.833 0 0 INIT_PERCAP 3.000 AMORT NONE
FOR 60 GROUP "3B" PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 52 "Arm2281" WL 00 WAC 6.375 (396000.00 / 396000.00); 396000.00
.50 .50 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO 6.100 25 6
SYNC_INT 12.375 1.000 6.375 0 0 INIT_PERCAP 3.000 AMORT NONE
FOR 60 GROUP "3B" PREPAY_FLAG YM FOR 18 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 53 "Arm2282" WL 00 WAC 6.445 (102545410.30 / 102545410.30);
102545410.30 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.505 24 6 SYNC_INT 12.445 1.000 6.445 0 0 INIT_PERCAP 3.000
AMORT NONE FOR 60 GROUP "3B" PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 54 "Arm2282" WL 00 WAC 6.250 (439000.00 / 439000.00); 439000.00
.50 .50 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO 5.500 25 6
SYNC_INT 12.250 1.000 6.250 0 0 INIT_PERCAP 3.000 AMORT NONE
FOR 60 GROUP "3B" PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 55 "Arm3273" WL 00 WAC 6.468 (6838524.00 / 6838524.00);
6838524.00 .50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.474 36 6 SYNC_INT 12.468 1.000 6.468 0 0 INIT_PERCAP 3.000
AMORT NONE FOR 60 GROUP "3B" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 56 "Arm5253" WL 00 WAC 6.286 (1902310.00 / 1902310.00);
1902310.00 .50 .50 359:1 359:1 360 NO_CHECK ARM CMT_1YR
5.500 60 12 SYNC_INT 12.286 2.000 6.286 0 0 AMORT
NONE FOR 60 GROUP "3B" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER
M 57 "Arm5253" WL 00 WAC 6.286 (0.02 / 0.02); 0.02
.50 .50 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO 5.474 36 6
SYNC_INT 12.468 1.000 6.468 0 0 INIT_PERCAP 3.000 AMORT NONE
FOR 60 GROUP 1 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_6MONTH95100); TEASER

Assumptions

100% PPC
Severity
Lag
Triggers Failing
FWD LI + 100

28% CPR
40%
12

M1

MATURITY

PPC	CDR Break-Even	Cum Loss Collat
80%	21%	19.13%
100%	23%	17.65%

M2

MATURITY

PPC	CDR Break-Even	Cum Loss Collat
80%	15%	15.51%
100%	15%	13.20%

M3

MATURITY

PPC	CDR Break-Even	Cum Loss Collat
80%	13%	14.08%
100%	14%	12.55%

M4

MATURITY

PPC	CDR Break-Even	Cum Loss Collat
80%	11%	12.50%
100%	12%	11.18%

M5

MATURITY

PPC	CDR Break-Even	Cum Loss Collat
80%	10%	11.66%
100%	10%	9.70%

M1

CALL

PPC	CDR Break-Even	Cum Loss Collat
80%	28%	17.06%
100%	32%	15.83%

M2

CALL

PPC	CDR Break-Even	Cum Loss Collat
80%	18%	13.06%
100%	20%	11.97%

M3

CALL

PPC	CDR Break-Even	Cum Loss Collat
80%	16%	12.11%
100%	18%	11.08%

M4

CALL

PPC	CDR Break-Even	Cum Loss Collat
80%	13%	10.61%
100%	14%	9.27%

M5

CALL

PPC	CDR Break-Even	Cum Loss Collat
80%	12%	10.01%
100%	13%	8.67%