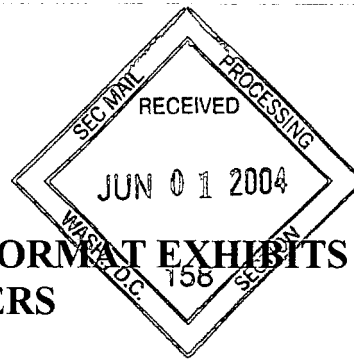


FORM SE
FORM FOR SUBMISSION OF PAPER FORMAT EXHIBITS
BY ELECTRONIC FILERS



Ameriquest Mortgage Securities Inc.
Exact Name of Registrant as Specified in Charter
Form 8-K, May 28, 2004, Series 2004-R5

0001102913
Registrant CIK Number
333-112203

Name of Person Filing the Document
(If Other than the Registrant)



PROCESSED
JUN 03 2004
THOMSON
FINANCIAL

A handwritten signature in dark ink, appearing to be 'W. Khan' or similar, located in the bottom right corner of the page.

SIGNATURES

Pursuant to the requirements of the Securities Exchange Act of 1934, the registrant has duly caused this report to be signed on its behalf by the undersigned thereunto duly authorized.

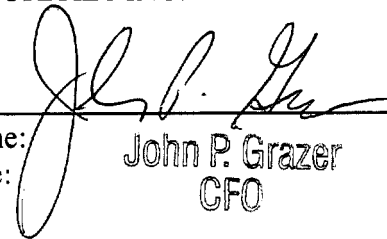
Dated: May 28, 2004

AMERIQUEST MORTGAGE
SECURITIES INC.

By: _____

Name: _____

Title: _____


John P. Grazer
CFO

**IN ACCORDANCE WITH RULE 202 OF REGULATION S-T, THIS EXHIBIT IS
BEING FILED IN PAPER PURSUANT TO A CONTINUING HARDSHIP EXEMPTION.**

EXHIBIT INDEX

<u>Exhibit No.</u>	<u>Description</u>	<u>Format</u>
99.1	Computational Materials	P*

* The Computational Materials have been filed on paper pursuant to a continuing hardship exemption from certain electronic requirements.

! AMQ04R05.CDI #CMOVER_3.0D ASSET_BACKED_HOMEEQUITY ! MAX_CF_VECTSIZE 620

!

!! Created by Intex Deal Maker v3.7.000 , subroutines 3.1

!! 05/19/2004 7:22 PM

!

DEAL_COMMENT _

"The tables and other statistical analyses (the 'Hypothetical Performance Data') that you will produce using Intex with the attached information are privileged and intended solely for use by you (the party to whom CSFB LLC provided the computer model used to generate them). The Hypothetical Performance Data will be generated by you using a computer model prepared by CSFB LLC in reliance upon information furnished by the issuer of the securities and its affiliates, the accuracy and completeness of which has not been verified by CSFB LLC or any other person. The computer model that you will use to prepare the Hypothetical Performance Data was furnished to you solely by CSFB LLC and not by the issuer of the securities. It may not be (a) used for any purpose other than to make a preliminary evaluation of the referenced securities or (b) provided by you to any third party other than your legal, tax, financial and/or accounting advisors for the purposes of evaluating the Hypothetical Performance Data. You agree that the Hypothetical Performance Data will be generated by or on behalf of you, and that neither CSFB LLC nor anyone acting on its behalf has generated or is in any way responsible for any Hypothetical Performance Data. ; _

! _

Numerous assumptions were used in preparing the computer model you will use to generate the Hypothetical Performance Data. Those assumptions may or may not be reflected in the Hypothetical Performance Data. As such, no assurance can be given as to the Hypothetical Performance Data's accuracy, appropriateness or completeness in any particular context; nor as to whether the Hypothetical Performance Data and/or the assumptions upon which it is based reflect present market conditions of future market performance. The Hypothetical Performance Data should not be construed as either projections or predictions or as legal, tax, financial or accounting advice. ; _

! _

Any weighted average lives, yields and principal payment periods shown in the Hypothetical Performance Data will be based on prepayment assumptions, and changes in such prepayment assumptions may dramatically affect such weighted average lives, yields and principal payment periods. In addition, it is possible that prepayments on the underlying assets will occur at rates slower or faster than the rates shown in the Hypothetical Performance Data. Furthermore, unless otherwise provided, the Hypothetical Performance Data assumes no losses on the underlying assets and no interest shortfall. The specific characteristics of the securities may differ from those shown in the Hypothetical Performance Data due to, among other things, differences between (a) the actual underlying assets and the hypothetical underlying assets used in preparing the Hypothetical Performance Data and (b) the assumptions used by you in producing the Hypothetical Performance Data and the actual assumptions used in pricing the actual securities. The principal amount, designation and terms of any security described in the Hypothetical Performance Data are subject to change prior to issuance. You should contact the CSFB LLC Trading Desk at (212-325-8549) to confirm the final principal amount, designation and terms of any security described in this communication prior to committing to purchase that security. Neither CSFB LLC nor any of its affiliates makes any representation or warranty as to the actual rate or timing of payments on any of the underlying assets or payments or yield on the securities.; _

! _

Although a registration statement (including a prospectus) relating to the securities discussed in this communication has been filed with the Securities

and Exchange Commission and is effective, the final prospectus supplement relating to the securities discussed in this communication has not yet been filed with the securities and Exchange Commission. This communication shall not constitute an offer to sell or the solicitation of an offer to buy nor shall there be any sale of the securities discussed in this communication in any state in which such offer, solicitation or sale would be unlawful prior to registration or qualification of such securities under the securities laws of any such state. The principal amount, designation and terms of any security described in the computer model and Hypothetical Performance Data are preliminary and subject to change prior to issuance. ; _

;- Prospective purchasers are referred to the final prospectus supplement relating to the securities discussed in this communication for definitive yield and maturity information regarding those securities, based on the final principal amounts, designations and terms of those securities. Once available, a final prospectus supplement may be obtained by contacting the CSFB LLC Trading Desk at (212-325-8549).; _

;- The computer model referenced herein supersedes all computer models related to the subject securities that have been made available to you previously. In addition, this computer model will be superseded in its entirety by the final prospectus supplement relating to the actual securities preliminarily described by this computer model.; _

;- Please be advised that the securities described herein may not be appropriate for all investors. Potential investors must be willing to assume, among other things, market price volatility, prepayment, yield curve and interest rate risks. Investors should make every effort to consider the risks of these securities. ; _

;- If you have received this communication in error, please notify the sending party immediately by telephone and return the original to such party by mail."

!
! Modeled in the Intex CMO Modeling Language, (WNYC1456761)
! 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 = "1F" "1A"
GROUP 2 = "2F" "2A"

!
DEFINE PREPAY PPC GROUP "1F" RISE_PERS 10 START_CPR 2 END_CPR 20
DEFINE PREPAY PPC GROUP "1A" RISE_PERS 1 START_CPR 27 END_CPR 27
DEFINE PREPAY PPC GROUP "2F" RISE_PERS 10 START_CPR 2 END_CPR 20
DEFINE PREPAY PPC GROUP "2A" RISE_PERS 1 START_CPR 27 END_CPR 27
!

!
DEFINE CONSTANT #OrigCollBal = 1000000409.03
DEFINE CONSTANT #OrigCollBal1 = 827644425.36
DEFINE CONSTANT #OrigCollBal1F = 167551210.44
DEFINE CONSTANT #OrigCollBal1A = 660093214.92
DEFINE CONSTANT #OrigCollBal2 = 172355983.67
DEFINE CONSTANT #OrigCollBal2F = 32502112.92
DEFINE CONSTANT #OrigCollBal2A = 139853870.75
!

DEFINE CONSTANT #OrigBondBal = 985000000.00
DEFINE CONSTANT #OrigBondBal1 = 827644425.36

```

DEFINE CONSTANT #OrigBondBal2 = 172355983.67
!
DEFINE CONSTANT #SpecSenEnhPct = 33.8%
DEFINE CONSTANT #AAATargPct = 66.200000000000%
DEFINE CONSTANT #AATargPct = 77.700000000000%
DEFINE CONSTANT #ATargPct = 87.200000000000%
DEFINE CONSTANT #AMTargPct = 89.200000000000%
DEFINE CONSTANT #BBBPTargPct = 91.700000000000%
DEFINE CONSTANT #BBBTargPct = 93.500000000000%
DEFINE CONSTANT #BBBMtargPct = 95.000000000000%
DEFINE CONSTANT #BBPTargPct = 97.000000000000%
DEFINE #BondBal = 985000000.00
DEFINE #BondBal1 = 687770000.00
DEFINE #BondBal2 = 143230000.00
!
FULL_DEALNAME:      AMQ04R05
!
DEAL SIZE:          $ 985000000.00
PRICING SPEED:      GROUP "1F" PPC 100%
PRICING SPEED:      GROUP "1A" PPC 100%
PRICING SPEED:      GROUP "2F" PPC 100%
PRICING SPEED:      GROUP "2A" PPC 100%
!  ISSUE DATE:       20040601
!  SETTLEMENT DATE:  20040604
!
Record date delay:  24
!
DEFINE TR_INDEXDEPS_ALL
!
DEFINE VARNAMEs #TrigDelinqFrac, #CumLossShft, #TrigCumLossFrac
!
DEAL_CLOCK_INFO _
    ISSUE_CDU_DATE      20040601 _
    DEAL_FIRSTPAY_DATE  20040725 _
!
!
DEFINE TABLE "CapNotional" (36, 2) = "CURDATE" "Balance"
    20040725.1      619,000,000.00
    20040825.1      602,513,878.68
    20040925.1      586,185,543.49
    20041025.1      570,003,545.74
    20041125.1      553,957,709.29
    20041225.1      538,039,461.19
    20050125.1      522,241,854.17
    20050225.1      506,559,579.43
    20050325.1      490,988,969.11
    20050425.1      475,536,670.56
    20050525.1      460,473,547.48
    20050625.1      445,789,738.02
    20050725.1      431,475,631.61
    20050825.1      417,521,862.50
    20050925.1      403,919,303.56
    20051025.1      390,659,060.12
    20051125.1      377,732,464.05
    20051225.1      365,131,067.93
    20060125.1      352,846,639.46
    20060225.1      340,871,155.91

```

20060325.1	329,196,798.78
20060425.1	317,815,948.55
20060525.1	306,721,176.59
20060625.1	295,905,257.73
20060725.1	285,370,814.50
20060825.1	275,100,803.62
20060925.1	265,088,544.98
20061025.1	255,327,528.24
20061125.1	245,811,408.46
20061225.1	236,534,001.94
20070125.1	227,489,282.05
20070225.1	218,671,375.23
20070325.1	210,074,557.11
20070425.1	201,693,248.65
20070525.1	193,522,012.49
20070625.1	0.00

!

DEFINE TABLE "CapRateSch" (36, 2) = "CURDATE" "Rate"

20040725.1	99
20040825.1	6.41
20040925.1	6.41
20041025.1	6.63
20041125.1	6.41
20041225.1	6.63
20050125.1	6.41
20050225.1	6.40
20050325.1	7.12
20050425.1	6.40
20050525.1	6.63
20050625.1	6.40
20050725.1	6.62
20050825.1	6.40
20050925.1	6.40
20051025.1	6.62
20051125.1	6.40
20051225.1	6.62
20060125.1	6.40
20060225.1	6.40
20060325.1	7.11
20060425.1	6.40
20060525.1	6.62
20060625.1	7.86
20060725.1	8.13
20060825.1	7.85
20060925.1	7.85
20061025.1	8.12
20061125.1	7.84
20061225.1	8.86
20070125.1	8.56
20070225.1	8.55
20070325.1	8.97
20070425.1	8.55
20070525.1	8.84
20070625.1	0.00

!

DEFINE DYNAMIC #CapBal = LOOKUP_TBL("STEP", Curdate, "CapNotional",
"CURDATE", "Balance")

```

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

```

```

  DEFINE DYNAMIC #CapRate = LOOKUP_TBL( "STEP", Curdate, "CapRateSch",
"CURDATE", "Rate" )
!

```

```

DEFINE TABLE "Cap2Notional" (30, 2) = "CURDATE" "Balance"

```

20040725.1	68,770,000.00
20040825.1	66,938,415.89
20040925.1	65,124,361.59
20041025.1	63,326,565.17
20041125.1	61,543,896.07
20041225.1	59,775,401.85
20050125.1	58,020,310.68
20050225.1	56,278,032.76
20050325.1	54,548,160.59
20050425.1	52,831,432.69
20050525.1	51,157,941.62
20050625.1	49,526,591.73
20050725.1	47,936,315.32
20050825.1	46,386,071.86
20050925.1	44,874,847.34
20051025.1	43,401,653.58
20051125.1	41,965,527.55
20051225.1	40,565,530.76
20060125.1	39,200,748.62
20060225.1	37,870,289.81
20060325.1	36,573,285.71
20060425.1	35,308,889.79
20060525.1	34,076,276.76
20060625.1	32,874,643.90
20060725.1	31,704,282.57
20060825.1	30,563,299.30
20060925.1	29,450,951.92
20061025.1	28,366,517.15
20061125.1	27,309,290.08
20061225.1	0.00

```

!
DEFINE TABLE "Cap2RateSch" (30, 2) = "CURDATE" "Rate"

```

20040725.1	99
20040825.1	6.31
20040925.1	6.31
20041025.1	6.53
20041125.1	6.31
20041225.1	6.53
20050125.1	6.31
20050225.1	6.30
20050325.1	7.02
20050425.1	6.30
20050525.1	6.53
20050625.1	6.30
20050725.1	6.52
20050825.1	6.30
20050925.1	6.30
20051025.1	6.52

20051125.1	6.30
20051225.1	6.52
20060125.1	6.30
20060225.1	6.30
20060325.1	7.01
20060425.1	6.30
20060525.1	6.52
20060625.1	7.76
20060725.1	8.03
20060825.1	7.75
20060925.1	7.75
20061025.1	8.02
20061125.1	7.74
20061225.1	0.00

!

```

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

```

!

```

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

```

!

```

DEFINE DYNAMIC #CapRate2 = LOOKUP_TBL( "STEP", Curdate, "Cap2RateSch",
"CURDATE", "Rate" )

```

!

```

DEFINE TABLE "Cap3Notional" (30, 2) = "CURDATE" "Balance"

```

20040725.1	143,230,000.00
20040825.1	139,360,472.80
20040925.1	135,533,514.90
20041025.1	131,746,422.26
20041125.1	127,996,764.95
20041225.1	124,282,464.12
20050125.1	120,601,796.81
20050225.1	116,953,398.65
20050325.1	113,336,264.39
20050425.1	109,749,746.11
20050525.1	106,253,979.13
20050625.1	102,846,654.59
20050725.1	99,525,522.62
20050825.1	96,288,390.88
20050925.1	93,133,123.05
20051025.1	90,057,637.40
20051125.1	87,059,905.43
20051225.1	84,137,950.42
20060125.1	81,289,846.19
20060225.1	78,513,715.74
20060325.1	75,807,730.03
20060425.1	73,170,106.73
20060525.1	70,599,109.02
20060625.1	68,093,044.43
20060725.1	65,654,035.72
20060825.1	63,276,513.62
20060925.1	60,958,919.13
20061025.1	58,699,732.96
20061125.1	56,497,474.57
20061225.1	0.00

!

```
DEFINE TABLE "Cap3RateSch" (30, 2) = "CURDATE" "Rate"
```

20040725.1	99
20040825.1	6.20
20040925.1	6.19
20041025.1	6.41
20041125.1	6.19
20041225.1	6.40
20050125.1	6.19
20050225.1	6.19
20050325.1	6.87
20050425.1	6.19
20050525.1	6.40
20050625.1	6.19
20050725.1	6.40
20050825.1	6.19
20050925.1	6.18
20051025.1	6.40
20051125.1	6.18
20051225.1	6.40
20060125.1	6.18
20060225.1	6.18
20060325.1	6.87
20060425.1	6.18
20060525.1	6.41
20060625.1	7.67
20060725.1	7.93
20060825.1	7.67
20060925.1	7.66
20061025.1	7.92
20061125.1	7.66
20061225.1	0.00

!

```
DEFINE DYNAMIC #CapBal3 = LOOKUP_TBL( "STEP", Curdate, "Cap3Notional",  
"CURDATE", "Balance" )
```

!

```
DEFINE DYNAMIC #CapBal3End = LOOKUP_TBL( "STEP", Curdate + 30, "Cap3Notional",  
"CURDATE", "Balance" )
```

!

```
DEFINE DYNAMIC #CapRate3 = LOOKUP_TBL( "STEP", Curdate, "Cap3RateSch",  
"CURDATE", "Rate" )
```

!

!

```
DEFINE TABLE "Cap4Notional" (30, 2) = "CURDATE" "Balance"
```

20040725.1	154,000,000.00
20040825.1	154,000,000.00
20040925.1	154,000,000.00
20041025.1	154,000,000.00
20041125.1	154,000,000.00
20041225.1	154,000,000.00
20050125.1	154,000,000.00
20050225.1	154,000,000.00
20050325.1	154,000,000.00
20050425.1	154,000,000.00
20050525.1	154,000,000.00
20050625.1	154,000,000.00
20050725.1	154,000,000.00
20050825.1	154,000,000.00

20050925.1	154,000,000.00
20051025.1	154,000,000.00
20051125.1	154,000,000.00
20051225.1	154,000,000.00
20060125.1	154,000,000.00
20060225.1	154,000,000.00
20060325.1	154,000,000.00
20060425.1	154,000,000.00
20060525.1	154,000,000.00
20060625.1	154,000,000.00
20060725.1	154,000,000.00
20060825.1	154,000,000.00
20060925.1	154,000,000.00
20061025.1	154,000,000.00
20061125.1	154,000,000.00
20061225.1	0.00

!

DEFINE TABLE "Cap4RateSch" (30, 2) = "CURDATE" "Rate"

20040725.1	99
20040825.1	5.36
20040925.1	5.36
20041025.1	5.58
20041125.1	5.36
20041225.1	5.58
20050125.1	5.36
20050225.1	5.36
20050325.1	6.07
20050425.1	5.36
20050525.1	5.58
20050625.1	5.35
20050725.1	5.58
20050825.1	5.35
20050925.1	5.35
20051025.1	5.57
20051125.1	5.35
20051225.1	5.57
20060125.1	5.35
20060225.1	5.35
20060325.1	6.06
20060425.1	5.35
20060525.1	5.58
20060625.1	6.82
20060725.1	7.08
20060825.1	6.81
20060925.1	6.81
20061025.1	7.07
20061125.1	6.80
20061225.1	0.00

!

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

!

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

!

DEFINE DYNAMIC #CapRate4 = LOOKUP_TBL("STEP", Curdate, "Cap4RateSch", "CURDATE", "Rate")

```

!
!
DEFINE #FloorCollat      = 0.50% * #OrigCollBal
DEFINE #SpecOCTarg       = 1.50% * #OrigCollBal
DEFINE STANDARDIZE OC_ACTUAL_VAL      #OC      = 15000409.03
DEFINE STANDARDIZE OCT_INITVAL        CONSTANT #InitOCTarg      = 1.50% *
#OrigCollBal
DEFINE STANDARDIZE OCT_STEPDOWN_MONTH CONSTANT #StepDownDate    = 37
DEFINE STANDARDIZE OCT_STEPDOWN_FRAC  CONSTANT #StepOCFrac      = 0.03
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 #NetRate1F = ( COLL_I_MISC("COUPON","1F") ) /
COLL_PREV_BAL("1F") * 1200
DEFINE DYNAMIC STICKY #NetRate1A = ( COLL_I_MISC("COUPON","1A") ) /
COLL_PREV_BAL("1A") * 1200
!
DEFINE DYNAMIC STICKY #NetRate2F = ( COLL_I_MISC("COUPON","2F") ) /
COLL_PREV_BAL("2F") * 1200
DEFINE DYNAMIC STICKY #NetRate2A = ( COLL_I_MISC("COUPON","2A") ) /
COLL_PREV_BAL("2A") * 1200
!
!
DEFINE DYNAMIC STICKY #NetRateActual360 = #Netrate * 30 / DAYS_DIFF(CURDATE ,
MONTHS_ADD(CURDATE,-1))
!
ifdef #cmover_3.0g _
DEFINE DYNAMIC STICKY #LifeCap = COLL_NET_LIFECAP
DEFINE DYNAMIC STICKY #LifeCap1 = COLL_NET_LIFECAP(1)
DEFINE DYNAMIC STICKY #LifeCap2 = COLL_NET_LIFECAP(2)
!
ifndef #cmover_3.0g _
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 1 #LifeCap1 = 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 TABLE "OC_CUMLOSS0" (4, 2) = "MONTH" "OC_CUMLOSS_FRAC0"
48.1 0.0275
60.1 0.0425
72.1 0.055
360.1 0.06

```

```

!
!
!
TOLERANCE WRITEDOWN_0LOSS 1.00
!
DEFINE TRANCHE "CAP_IN", "CAP_IN2", "CAP_IN3", "CAP_IN4", "A1A", "A1B", "A2",
"A3", "A4", "M1", "M2", "M3", "M4", "M5", "M6", "M7", "R", "R_PP"
!
DEFINE DYNAMIC STICKY #Scaling_Factor = IF CURMONTH EQ 1 THEN 30 / 51 ELSE 30 /
DAYS_DIFF(CURDATE, MONTHS_ADD(CURDATE, -1))
!
!!Define Net WAC Caps
DEFINE DYNAMIC STICKY #NetWac1 = (COLL_I_MISC("COUPON",1)) / COLL_PREV_BAL(1) *
1200
DEFINE DYNAMIC STICKY #NetWac2 = (COLL_I_MISC("COUPON",2)) / COLL_PREV_BAL(2) *
1200
!
DEFINE DYNAMIC STICKY #NetWac1_Adj = #NetWac1 * #Scaling_Factor
DEFINE DYNAMIC STICKY #NetWac2_Adj = #NetWac2 * #Scaling_Factor
!
!
!
!!DEFINE SUBORDINATE BOND BALANCE
DEFINE DYNAMIC STICKY #SUBBAL1 = COLL_PREV_BAL(1) - BBAL("A1A", "A1B")
DEFINE DYNAMIC STICKY #SUBBAL2 = COLL_PREV_BAL(2) - BBAL("A2", "A3", "A4")
DEFINE DYNAMIC STICKY #SUBBAL = #SUBBAL1 + #SUBBAL2
!
DEFINE DYNAMIC STICKY #SubWacCap1 = ((#NetWac1 * #SUBBAL1 / #SUBBAL) +
(#NetWac2 * #SUBBAL2 / #SUBBAL))
DEFINE DYNAMIC STICKY #SubWacCap = IF CURMONTH EQ 1 THEN #SubWacCap1 * (30/51)
ELSE #SubWacCap1
DEFINE DYNAMIC STICKY #SubWacCap_Adj = #SubWacCap * #Scaling_Factor
!
!
!
!!DEFINE MAXIMUM LIFE CAPS
DEFINE DYNAMIC STICKY #MaxLifeCap1 = (#LifeCap1) + (OPTIMAL_INTPMT("CAP_IN") +
OPTIMAL_INTPMT("CAP_IN2")) / BBAL("A1A", "A1B") * 1200
DEFINE DYNAMIC STICKY #MaxLifeCap2 = (#LifeCap2) + OPTIMAL_INTPMT("CAP_IN3") /
BBAL("A2", "A3", "A4") * 1200
DEFINE DYNAMIC STICKY #MaxLifeCap3 = ((#LifeCap1 * #SUBBAL1 / #SUBBAL) +
(#LifeCap2 * #SUBBAL2 / #SUBBAL)) + OPTIMAL_INTPMT("CAP_IN4") / BBAL("M1", "M2",
"M3", "M4", "M5", "M6", "M7") * 1200
!
!
!!Define Effective Caps
DEFINE DYNAMIC STICKY #A1A_YM_INCOME = (OPTIMAL_INTPMT("CAP_IN") /
BBAL("A1A") * 1200)
DEFINE DYNAMIC STICKY #A1B_YM_INCOME = (OPTIMAL_INTPMT("CAP_IN2") /
BBAL("A1B") * 1200)
DEFINE DYNAMIC STICKY #ANC_YM_INCOME = (OPTIMAL_INTPMT("CAP_IN3") /
BBAL("A2", "A3", "A4") * 1200)
DEFINE DYNAMIC STICKY #SUB_YM_INCOME = (OPTIMAL_INTPMT("CAP_IN4") /
BBAL("M1", "M2", "M3", "M4", "M5", "M6", "M7") * 1200)
!
DEFINE DYNAMIC #A1A_Effective_Cap = #NetWac1_Adj + (#A1A_YM_INCOME *
#Scaling_Factor)

```

```

DEFINE DYNAMIC #A1B_Effective_Cap      = #NetWac1_Adj + (#A1B_YM_INCOME *
#Scaling_Factor)
DEFINE DYNAMIC #ANC_Effective_Cap      = #NetWac2_Adj + (#ANC_YM_INCOME *
#Scaling_Factor)
DEFINE DYNAMIC #SUB_Effective_Cap      = #SubWacCap_Adj + (#SUB_YM_INCOME *
#Scaling_Factor)
    INITIAL INDEX    LIBOR_6MO          1.53
    INITIAL INDEX    LIBOR_1MO          1.10
!
!!DEFINE TRANCHE "CAP_IN", "CAP_IN2", "CAP_IN3", "CAP_IN4", "A1A", "A1B", "A2",
"A3", "A4", "M1", "M2", "M3", "M4", "M5", "M6", "M7", "R", "R_PP"
!
!
Tranche "CAP_IN" PSEUDO HEDGE
    Block $ 619,000,000.00 at 0.00 FLOAT NOTIONAL WITH FORMULA BEGIN ( #CapBal );
END ( #CapBalEnd );
    DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
    Delay 0 Dated 20040604 Next 20040725
    ((1 * MIN(8.97, LIBOR_1MO)) + (-1 * #CapRate))
    0      999
!
Tranche "CAP_IN2" PSEUDO HEDGE
    Block $ 68,770,000.00 at 0.00 FLOAT NOTIONAL WITH FORMULA BEGIN ( #CapBal2 );
END ( #CapBal2End );
    DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
    Delay 0 Dated 20040604 Next 20040725
    ((1 * MIN(8.12, LIBOR_1MO)) + (-1 * #CapRate2))
    0      999
!
Tranche "CAP_IN3" PSEUDO HEDGE
    Block $ 143,230,000.00 at 0.00 FLOAT NOTIONAL WITH FORMULA BEGIN ( #CapBal3
); END ( #CapBal3End );
    DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
    Delay 0 Dated 20040604 Next 20040725
    ((1 * MIN(8.28, LIBOR_1MO)) + (-1 * #CapRate3))
    0      999
!
Tranche "CAP_IN4" PSEUDO HEDGE
    Block $ 154,000,000.00 at 0.00 FLOAT NOTIONAL WITH FORMULA BEGIN ( #CapBal4
); END ( #CapBal4End );
    DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
    Delay 0 Dated 20040604 Next 20040725
    ((1 * MIN(7.47, LIBOR_1MO)) + (-1 * #CapRate4))
    0      999
!
Tranche "A1A" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
    Block 619000000.00 at 1.38 GROUP 1 FREQ M FLOAT RESET M _
    COUPONCAP 30360 NONE ( #NetRate1 );
    DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
    Delay 0 Dated 20040604 Next 20040725
    ( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 0.56 ELSE 0.28 )) , #MaxLifeCap1 * 30 / NDAYS_ACCRUE_INT("A1A#1")) )
    0      999
!
Tranche "A1B" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
    Block 68770000.00 at 1.48 GROUP 1 FREQ M FLOAT RESET M _
    COUPONCAP 30360 NONE ( #NetRate1 );

```

```

DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
Delay 0 Dated 20040604 Next 20040725
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 0.76 ELSE 0.38 )), #MaxLifeCap1 * 30 / NDAYS_ACCRUE_INT("A1B#1"))) )
0 999

```

```

!
Tranche "A2" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 76200000.00 at 1.21 GROUP 2 FREQ M FLOAT RESET M _
COUPONCAP 30360 NONE ( #NetRate2 ); _
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
Delay 0 Dated 20040604 Next 20040725
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 0.22 ELSE 0.11 )), #MaxLifeCap2 * 30 / NDAYS_ACCRUE_INT("A2#1"))) )
0 999

```

```

!
Tranche "A3" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 37800000.00 at 1.34 GROUP 2 FREQ M FLOAT RESET M _
COUPONCAP 30360 NONE ( #NetRate2 ); _
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
Delay 0 Dated 20040604 Next 20040725
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 0.48 ELSE 0.24 )), #MaxLifeCap2 * 30 / NDAYS_ACCRUE_INT("A3#1"))) )
0 999

```

```

!
Tranche "A4" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 29230000.00 at 1.6 GROUP 2 FREQ M FLOAT RESET M _
COUPONCAP 30360 NONE ( #NetRate2 ); _
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
Delay 0 Dated 20040604 Next 20040725
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 1. ELSE 0.5 )), #MaxLifeCap2 * 30 / NDAYS_ACCRUE_INT("A4#1"))) )
0 999

```

```

!
Tranche "M1" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 57500000.00 at 1.67 FREQ M FLOAT RESET M _
COUPONCAP 30360 NONE ( #SubWacCap ); _
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
Delay 0 Dated 20040604 Next 20040725
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 0.855 ELSE 0.57 )), #MaxLifeCap3 * 30 / NDAYS_ACCRUE_INT("M1#1"))) )
0 999

```

```

!
Tranche "M2" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 47500000.00 at 2.3 FREQ M FLOAT RESET M _
COUPONCAP 30360 NONE ( #SubWacCap ); _
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
Delay 0 Dated 20040604 Next 20040725
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 1.8 ELSE 1.2 )), #MaxLifeCap3 * 30 / NDAYS_ACCRUE_INT("M2#1"))) )
0 999

```

```

!
Tranche "M3" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 10000000.00 at 2.55 FREQ M FLOAT RESET M _
COUPONCAP 30360 NONE ( #SubWacCap ); _
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
Delay 0 Dated 20040604 Next 20040725

```

```
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 2.175 ELSE 1.45 )), #MaxLifeCap3 * 30 / NDAYS_ACCRUE_INT("M3#1"))) )
0      999
```

!

Tranche "M4" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);

Block 12500000.00 at 3.37 FREQ M FLOAT RESET M _

COUPONCAP 30360 NONE (#SubWacCap); _

DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _

Delay 0 Dated 20040604 Next 20040725

```
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 3.405 ELSE 2.27 )), #MaxLifeCap3 * 30 / NDAYS_ACCRUE_INT("M4#1"))) )
0      999
```

!

Tranche "M5" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);

Block 90000000.00 at 3.45 FREQ M FLOAT RESET M _

COUPONCAP 30360 NONE (#SubWacCap); _

DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _

Delay 0 Dated 20040604 Next 20040725

```
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 3.525 ELSE 2.35 )), #MaxLifeCap3 * 30 / NDAYS_ACCRUE_INT("M5#1"))) )
0      999
```

!

Tranche "M6" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);

Block 75000000.00 at 5.6 FREQ M FLOAT RESET M _

COUPONCAP 30360 NONE (#SubWacCap); _

DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _

Delay 0 Dated 20040604 Next 20040725

```
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 6.75 ELSE 4.5 )), #MaxLifeCap3 * 30 / NDAYS_ACCRUE_INT("M6#1"))) )
0      999
```

!

Tranche "M7" JUN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);

Block 100000000.00 at 2.1 FREQ M FLOAT RESET M _

COUPONCAP 30360 NONE (#SubWacCap); _

DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _

Delay 0 Dated 20040604 Next 20040725

```
( MIN((1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 1.5 ELSE 1 )), #MaxLifeCap3 * 30 / NDAYS_ACCRUE_INT("M7#1"))) )
0      999
```

!

Tranche "R" SEN_RES

Block 1000000409.03 at 0 NOTIONAL WITH GROUP 0 SURPLUS _

DAYCOUNT 30360 BUSINESS_DAY NONE _

FREQ M Delay 24 Dated 20040601 Next 20040725

!

Tranche "R_PP" SEN_PEN_NO

Block 1000000409.03 at 0 NOTIONAL WITH GROUP 0 _

DAYCOUNT 30360 BUSINESS_DAY NONE _

FREQ M Delay 24 Dated 20040601 Next 20040725

!

Tranche "SUBS" PSEUDO

Block USE PCT 100.0 100.0 of M1#1

Block USE PCT 100.0 100.0 of M2#1

Block USE PCT 100.0 100.0 of M3#1

Block USE PCT 100.0 100.0 of M4#1

Block USE PCT 100.0 100.0 of M5#1

Block USE PCT 100.0 100.0 of M6#1

```

Block    USE PCT 100.0 100.0 of M7#1
!
Tranche "#NetRate"          SYMVAR
!
Tranche "#NetRateActual360"  SYMVAR
!
Tranche "#NetRate1F"        SYMVAR
!
Tranche "#NetRate1A"        SYMVAR
!
Tranche "#NetRate2F"        SYMVAR
!
Tranche "#NetRate2A"        SYMVAR
!
Tranche "#OC"               SYMVAR
Tranche "#SpecOCTarg"       SYMVAR
!
Tranche "DEAL_PLUGIN" PSEUDO
  Block USE PCT 100.0 100.0 OF "A1A#1"
  Block USE PCT 100.0 100.0 OF "A1B#1"
  Block USE PCT 100.0 100.0 OF "A2#1"
  Block USE PCT 100.0 100.0 OF "A3#1"
  Block USE PCT 100.0 100.0 OF "A4#1"
  Block USE PCT 100.0 100.0 OF "M1#1"
  Block USE PCT 100.0 100.0 OF "M2#1"
  Block USE PCT 100.0 100.0 OF "M3#1"
  Block USE PCT 100.0 100.0 OF "M4#1"
  Block USE PCT 100.0 100.0 OF "M5#1"
  Block USE PCT 100.0 100.0 OF "M6#1"
  Block USE PCT 100.0 100.0 OF "M7#1"
  Block USE PCT   0.0 100.0 OF "R#1"
  Block USE PCT   0.0 100.0 OF "R_PP#1"
!
!
DEFINE PSEUDO_TRANCHE COLLAT _
  Delay 24 Dated 20040601 Next 20040725 Settle 20040604
DEFINE PSEUDO_TRANCHE COLLAT GROUP 1 _
  Delay 24 Dated 20040601 Next 20040725 Settle 20040604
DEFINE PSEUDO_TRANCHE COLLAT GROUP 2 _
  Delay 24 Dated 20040601 Next 20040725 Settle 20040604
!
RESERVE_FUND "YmRsvFnd"    FUNDING_FROM RULES
!
HEDGE "CAP" _
      TYPE  CAP _
      LEG   "FLT"      DEAL_RECEIVES    OPTIMAL_INTPMT  "CAP_IN"
!
HEDGE "CAP2" _
      TYPE  CAP _
      LEG   "FLT"      DEAL_RECEIVES    OPTIMAL_INTPMT  "CAP_IN2"
!
HEDGE "CAP3" _
      TYPE  CAP _
      LEG   "FLT"      DEAL_RECEIVES    OPTIMAL_INTPMT  "CAP_IN3"
!
HEDGE "CAP4" _
      TYPE  CAP _

```

LEG "FLT" DEAL_RECEIVES OPTIMAL_INTPMT "CAP_IN4"

1

CLASS "A1A"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "A1A" _
CLASS "A1B"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "A1B" _
CLASS "A2"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "A2" _
CLASS "A3"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "A3" _
CLASS "A4"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "A4" _
CLASS "AA"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "M1" _
CLASS "A"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "M2" _
CLASS "A-"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "M3" _
CLASS "BBB+"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "M4" _
CLASS "BBB"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "M5" _
CLASS "BBB-"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "M6" _
CLASS "BB+"	NO_BUILD_TRANCHE _
	SHORTFALL_PAYBACK COUPONCAP TRUE _
	SHORTFALL_EARN_INT COUPONCAP TRUE _
	= "M7" _
CLASS "RESID"	= "R#1" "R_PP#1"
CLASS "AAA_1"	WRITEDOWN_BAL PRORATA ALLOCATION _
	= "A1A" "A1B"
CLASS "AAA_2"	WRITEDOWN_BAL PRORATA ALLOCATION _
	= "A2" "A3" "A4"
CLASS "AAA"	DISTRIB_CLASS PRORATA WRITEDOWN_BAL PRORATA ALLOCATION _

```

!
!
= "AAA_1" "AAA_2"
CLASS "ROOT" _
    WRITEDOWN_BAL RULES _
    DISTRIB_CLASS RULES _
    SHORTFALL_PAYBACK PRINCIPAL_LOSS TRUE _
    SHORTFALL_EARN_INT INTEREST TRUE _
        = "AAA" "AA" "A" "A-" "BBB+" "BBB" "BBB-" "BB+" "RESID"
!
!
    DEFINE PSEUDO_TRANCHE CLASS "AAA"                Delay 24   Dated 20040601   Next
20040725 DAYCOUNT 30360 BUSINESS_DAY NONE
!
    DEFINE PSEUDO_TRANCHE CLASS "AAA_1"              Delay 24   Dated 20040601   Next
20040725 DAYCOUNT 30360 BUSINESS_DAY NONE
!
    DEFINE PSEUDO_TRANCHE CLASS "AAA_2"              Delay 24   Dated 20040601   Next
20040725 DAYCOUNT 30360 BUSINESS_DAY NONE
!
!
CROSSOVER When 0
!
!
TRIGGER "StepUp-DlqRatio" _
    FULL_NAME      "Step Up Delinquency Trigger" _
    ORIG_TESTVAL   0.000% _
    TESTVAL        ( #TrigDelinqFrac); _
    ORIG_TARGETVAL 2% _
    TARGETVAL       (2%); _
    TRIGVAL         LODIFF
!
!
TRIGGER "StepUp-CumLoss" _
    FULL_NAME      "Step Up Cumulative Loss Trigger" _
    ORIG_TESTVAL   0.000% _
    TESTVAL        ( #TrigCumLossFrac); _
    ORIG_TARGETVAL 2.75% _
    TARGETVAL       (#CumLossShft); _
    TRIGVAL         LODIFF
!
!
TRIGGER "STEPUP_TRIGGER" _
    FULL_NAME      "Step Up Trigger" _
    DEFINITION     "A Step Up Trigger exists, if_
; (1) the quotient of (A) the aggregate principal balance of all_
mortgage loans 30 or more days delinquent, averaged over previous 1 months _
and (B) the principal balance of the loans, exceeds 2%. _
or; (2) 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 <=          %; _
                        48             2.75%; _
                        60             4.25%; _
                        72             5.50%; _
                       360            6.00%; _
"
-
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 not occurred,
```

or % of the balance when the trigger first occurred if a stepdown has occurred."

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

!

```
OPTIONAL REDEMPTION: "Cleanup"  
COLL_FRAC 10%  
PRICE_P ( COLL_BAL );
```

!

!

```
INTEREST_SHORTFALL GROUP 1 FULL_PREPAY Compensate Pro_rata_  
PARTIAL_PREPAY Compensate Pro_rata_  
LOSS NO_Compensate SUBORDINATED ACCUM
```

!

```
INTEREST_SHORTFALL GROUP 2 FULL_PREPAY Compensate Pro_rata_  
PARTIAL_PREPAY Compensate Pro_rata_  
LOSS NO_Compensate SUBORDINATED ACCUM
```

!

```
DEFINE MACRO BLOCK #AAA_Int =
```

```
{
```

```
-----  
from : CLASS ( "AAA" )  
pay : CLASS INTEREST PRO_RATA ( "AAA_1"; "AAA_2" )  
-----
```

```
-----  
from : CLASS ( "AAA_1" )  
pay : CLASS INTEREST PRO_RATA ( "A1A"; "A1B" )  
-----
```

```
-----  
from : CLASS ( "AAA_2" )  
pay : CLASS INTEREST PRO_RATA ( "A2"; "A3"; "A4" )  
-----
```

```
}
```

```
DEFINE MACRO BLOCK #AAA_InS =
```

```
{
```

```
-----  
from : CLASS ( "AAA" )  
pay : CLASS INTSHORT PRO_RATA ( "AAA_1"; "AAA_2" )  
-----
```

```
-----  
from : CLASS ( "AAA_1" )  
pay : CLASS INTSHORT PRO_RATA ( "A1A"; "A1B" )  
-----
```

```
-----  
from : CLASS ( "AAA_2" )  
pay : CLASS INTSHORT PRO_RATA ( "A2"; "A3"; "A4" )  
-----
```

```
}
```

```
DEFINE MACRO BLOCK #AAA_Prn[1] =
```

```
{
```

```
-----  
when : IS_TRUE( {#1} )  
calculate : #SeniorPrinc = #PrincPmt / #DistribAmt * #ClassAAAPDA  
calculate : #SeniorXtraP = #ClassAAAPDA - #SeniorPrinc
```

!

```
calculate : #SeniorPDA1 = MAX( 0, MIN( #ClassAAA_1PDA,  
#SenDistribAmt1 ) ) + _
```

```

MIN( #ClassAAA_1PDADefic,
#ClassAAA_1PDADefic/#TotalSenPDADefic * #TotalExcessDistrib)
calculate : #SeniorPDA2 = MAX( 0, MIN( #ClassAAA_2PDA,
#SenDistribAmt2 ) ) + _
MIN( #ClassAAA_2PDADefic,
#ClassAAA_2PDADefic/#TotalSenPDADefic * #TotalExcessDistrib)
-----
from : SUBACCOUNT ( #SeniorPDA1, CLASS "AAA" )
pay : CLASS BALANCE SEQUENTIAL ( "AAA_1" )
-----
from : SUBACCOUNT ( #SeniorPDA2, CLASS "AAA" )
pay : CLASS BALANCE SEQUENTIAL ( "AAA_2" )
-----
from : CLASS ( "AAA" )
pay : CLASS BALANCE PRO_RATA ( "AAA_1"; "AAA_2" )
-----
!
from : CLASS ( "AAA_1" )
pay : CLASS BALANCE PRO_RATA ( "A1A" ; "A1B" )
-----
!
from : CLASS ( "AAA_2" )
pay : CLASS BALANCE SEQUENTIAL ( "A2", "A3", "A4" )
-----
!
-----
from : CLASS ( "A1A" )
pay : SEQUENTIAL ( "A1A#1" )
-----
from : CLASS ( "A1B" )
pay : SEQUENTIAL ( "A1B#1" )
-----
from : CLASS ( "A2" )
pay : SEQUENTIAL ( "A2#1" )
-----
from : CLASS ( "A3" )
pay : SEQUENTIAL ( "A3#1" )
-----
from : CLASS ( "A4" )
pay : SEQUENTIAL ( "A4#1" )
-----
}
DEFINE MACRO BLOCK #AA_Prn =
{
-----
from : CLASS ( "AA" )
pay : SEQUENTIAL ( "M1#1" )
-----
}
DEFINE MACRO BLOCK #A_Prn =
{
-----
from : CLASS ( "A" )
pay : SEQUENTIAL ( "M2#1" )
-----
}
DEFINE MACRO BLOCK #AM_Prn =

```

```

{
-----
    from : CLASS ( "A-" )
    pay : SEQUENTIAL ( "M3#1" )
-----
}
DEFINE MACRO BLOCK #BBBP_Prn =
{
-----
    from : CLASS ( "BBB+" )
    pay : SEQUENTIAL ( "M4#1" )
-----
}
DEFINE MACRO BLOCK #BBB_Prn =
{
-----
    from : CLASS ( "BBB" )
    pay : SEQUENTIAL ( "M5#1" )
-----
}
DEFINE MACRO BLOCK #BBBM_Prn =
{
-----
    from : CLASS ( "BBB-" )
    pay : SEQUENTIAL ( "M6#1" )
-----
}
DEFINE MACRO BLOCK #BBP_Prn =
{
-----
    from : CLASS ( "BB+" )
    pay : SEQUENTIAL ( "M7#1" )
-----
}
!
!
CMO Block Payment Rules
-----
    from : CASH_ACCOUNT (100)
    subject to : CEILING ( (COLL("PREPAYPENALTY")) )
    pay : CREDIT_ENHANCEMENT ("YmRsvFnd")
-----
    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("A1A#1",
"A1B#1", "A2#1", "A3#1", "A4#1", "M1#1", "M2#1", "M3#1", "M4#1", "M5#1", "M6#1",
"M7#1") - #Princ))
!
    calculate : #XSSpread = MAX( 0, #Interest - ( COLL_YM * (100% -
0%)) - OPTIMAL_INTPMT("ROOT") - INTSHORT_ACCUM("ROOT") +
COUPONCAP_SHORTFALL("ROOT") )
!
    calculate : #FloorOCTotal = #FloorOCTarg
!
    calculate : #StepOCTarg = COLL_BAL * #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 - ( COLL_YM * (100% -
0%)) - OPTIMAL_INTPMT("ROOT") - INTSHORT_ACCUM("ROOT") + #OCSurplus +
COUPONCAP_SHORTFALL("ROOT"))
!
    calculate : #SubDefic = MAX ( 0, ( BBAL("ROOT") - #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 : #SenDistribAmt1 = #PrincPmt * #PrincFrac1 + (#DistribAmt -
#PrincPmt) * #XtraPFrac1
    calculate : #SenDistribAmt2 = #PrincPmt * #PrincFrac2 + (#DistribAmt -
#PrincPmt) * #XtraPFrac2
!
    calculate : #FloorOCTotal1 = #OrigCollBal1/#OrigCollBal *
#FloorOCTotal
    calculate : #FloorOCTotal2 = #OrigCollBal2/#OrigCollBal *
#FloorOCTotal
!
    calculate : #ClassAAA_1PDA = BBAL("A1A", "A1B") _

```

```

- MIN(COLL_BAL(1) - #FloorOCTotal1,
#AAATargPct * COLL_BAL(1))
  calculate : #ClassAAA_1PDA = MAX( 0.0, MIN(BBAL("A1A", "A1B"),
#ClassAAA_1PDA ))
  calculate : #ClassAAA_2PDA = BBAL("A2", "A3", "A4") -
- MIN(COLL_BAL(2) - #FloorOCTotal2,
#AAATargPct * COLL_BAL(2))
  calculate : #ClassAAA_2PDA = MAX( 0.0, MIN(BBAL("A2", "A3", "A4"),
#ClassAAA_2PDA ))
!
  calculate : #ClassAAA_1PDADefic = MAX( 0.0, #ClassAAA_1PDA -
#SenDistribAmt1 )
  calculate : #ClassAAA_2PDADefic = MAX( 0.0, #ClassAAA_2PDA -
#SenDistribAmt2 )
  calculate : #TotalSenPDADefic = #ClassAAA_1PDADefic + #ClassAAA_2PDADefic
!
  calculate : #ExcessDistrib1 = MAX( 0.0, #SenDistribAmt1 -
#ClassAAA_1PDA )
  calculate : #ExcessDistrib2 = MAX( 0.0, #SenDistribAmt2 -
#ClassAAA_2PDA )
  calculate : #TotalExcessDistrib = #ExcessDistrib1 + #ExcessDistrib2
!
  calculate : #ExcessBalance1 = BBAL("AAA_1") - MIN( #SenDistribAmt1,
#ClassAAA_1PDA )
  calculate : #ExcessBalance2 = BBAL("AAA_2") - MIN( #SenDistribAmt2,
#ClassAAA_2PDA )
  calculate : #ExcessBalance = #ExcessBalance1 + #ExcessBalance2
!
  calculate : #ExcessDistrib = 0
!
  calculate : #ClassAAAPDA = #ClassAAA_1PDA + #ClassAAA_2PDA +
#ExcessDistrib
  calculate : #ClassAAAPDA = MIN( #ClassAAAPDA, #DistribAmt )
!
!!!***** END OF SENIOR ENHANCEMENT PCT CALCULATION *****
!
!! calculate : #SenEnhancePct = (COLL_BAL - (BBAL("AAA") - #ClassAAAPDA
)) / COLL_BAL
  calculate : #SenEnhancePct = (COLL_PREV_BAL - BBAL("AAA") ) /
COLL_BAL
!
  calculate : #StepDownBal = (#SenEnhancePct - #SpecSenEnhPct) + 1E-8
GE 0.00
!
  calculate : #StepDown = #StepDown OR ( BBAL("AAA") LT 0.01 ) OR (
#StepDownDatePass AND #StepDownBal )
!
  calculate : #TrigDelinqFrac = AVG_COLL("RATE",-1,1,1)
!
  calculate : #CumLossShift = LOOKUP_TBL( "STEP", CURMONTH ,
"OC_CUMLOSS0", "MONTH", "OC_CUMLOSS_FRAC0" )
  calculate : #TrigCumLossFrac = DELINQ_LOSS_ACCUM / #OrigCollBal
!
  calculate : #TrigEvent = TRIGGER("STEPUP_TRIGGER")
!
  calculate : #TrigOCTargPre = #PrevSpecOC
!

```

```

    calculate : #TrigOCTargPost      = #PrevSpecOC
!
    calculate : #SpecOCTarg          = IF #StepDown _
    THEN IF #TrigEvent _
    THEN MAX( MIN( #InitOCTarg,
#StepOCTarg ) , #TrigOCTargPost, #FloorOCTotal ) _
    ELSE MAX( MIN( #InitOCTarg,
#StepOCTarg ) , #FloorOCTotal ) _
    ELSE IF #TrigEvent _
    THEN MAX( #InitOCTarg, #TrigOCTargPre,
#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") - INTSHORT_ACCUM("ROOT") + #OCSurplus +
COUPONCAP_SHORTFALL("ROOT"))
!
    calculate : #SubDefic            = MAX ( 0, ( BBAL("ROOT") - #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 : #SenDistribAmt1      = #PrincPmt * #PrincFrac1 + (#DistribAmt -
#PrincPmt) * #XtraPFrac1
    calculate : #SenDistribAmt2      = #PrincPmt * #PrincFrac2 + (#DistribAmt -
#PrincPmt) * #XtraPFrac2
!
    calculate : #FloorOCTotal1       = #OrigCollBal1/#OrigCollBal *
#FloorOCTotal
    calculate : #FloorOCTotal2       = #OrigCollBal2/#OrigCollBal *
#FloorOCTotal
!
    calculate : #ClassAAA_1PDA       = IF ( #TrigEvent OR ( #StepDown EQ 0.0 ) ) _
    THEN #SenDistribAmt1 _
    ELSE BBAL("A1A", "A1B") _
    - MIN(COLL_BAL(1) - #FloorOCTotal1,
#AAATargPct * COLL_BAL(1))
    calculate : #ClassAAA_1PDA       = MAX( 0.0, MIN(BBAL("A1A", "A1B"),
#ClassAAA_1PDA ) )

```

```

        calculate : #ClassAAA_2PDA = IF (#TrigEvent OR (#StepDown EQ 0.0)) _
                                THEN #SenDistribAmt2 _
                                ELSE BBAL("A2", "A3", "A4") _
                                - MIN(COLL_BAL(2) - #FloorOCTotal2,
#AAATargPct * COLL_BAL(2))
        calculate : #ClassAAA_2PDA = MAX( 0.0, MIN(BBAL("A2", "A3", "A4"),
#ClassAAA_2PDA ))
!
        calculate : #ClassAAA_1PDADefic = MAX( 0.0, #ClassAAA_1PDA -
#SenDistribAmt1 )
        calculate : #ClassAAA_2PDADefic = MAX( 0.0, #ClassAAA_2PDA -
#SenDistribAmt2 )
        calculate : #TotalSenPDADefic = #ClassAAA_1PDADefic + #ClassAAA_2PDADefic
!
        calculate : #ExcessDistrib1 = MAX( 0.0, #SenDistribAmt1 -
#ClassAAA_1PDA )
        calculate : #ExcessDistrib2 = MAX( 0.0, #SenDistribAmt2 -
#ClassAAA_2PDA )
        calculate : #TotalExcessDistrib = #ExcessDistrib1 + #ExcessDistrib2
!
        calculate : #ExcessBalance1 = BBAL("AAA_1") - MIN( #SenDistribAmt1,
#ClassAAA_1PDA )
        calculate : #ExcessBalance2 = BBAL("AAA_2") - MIN( #SenDistribAmt2,
#ClassAAA_2PDA )
        calculate : #ExcessBalance = #ExcessBalance1 + #ExcessBalance2
!
        calculate : #ExcessDistrib = IF (#TrigEvent OR (#StepDown EQ 0.0)) _
                                THEN MIN( #ExcessBalance,
#ExcessDistrib1 + #ExcessDistrib2 ) _
                                ELSE 0
!
        calculate : #ClassAAAPDA = #ClassAAA_1PDA + #ClassAAA_2PDA +
#ExcessDistrib
        calculate : #ClassAAAPDA = MIN( #ClassAAAPDA, #DistribAmt )
!
        calculate : #ClassAAPDA = IF (#TrigEvent OR (#StepDown EQ 0.0)) _
                                THEN #DistribAmt - #ClassAAAPDA _
                                ELSE BBAL("A1A", "A1B", "A2", "A3",
"A4", "M1") - #ClassAAAPDA _
                                - MIN(COLL_BAL - #FloorOCTotal,
#AAATargPct * COLL_BAL)
        calculate : #ClassAAPDA = MAX( 0.0, MIN(BBAL("M1"), #ClassAAPDA ))
        calculate : #ClassAAPDA = MAX( 0, MIN( #ClassAAPDA, #DistribAmt -
#ClassAAAPDA ) )
!
!
        calculate : #ClassAPDA = IF (#TrigEvent OR (#StepDown EQ 0.0)) _
                                THEN #DistribAmt - #ClassAAAPDA _
                                ELSE BBAL("A1A", "A1B", "A2", "A3",
"A4", "M1", "M2") - #ClassAAAPDA - #ClassAAPDA _
                                - MIN(COLL_BAL - #FloorOCTotal,
#ATargPct * COLL_BAL)
        calculate : #ClassAPDA = MAX( 0.0, MIN(BBAL("M2"), #ClassAPDA ))
        calculate : #ClassAPDA = MAX( 0, MIN( #ClassAPDA, #DistribAmt -
#ClassAAAPDA - #ClassAAPDA ) )
!

```

```

!
    calculate : #ClassAMPDA = IF (#TrigEvent OR (#StepDown EQ 0.0)) _
                                THEN #DistribAmt - #ClassAAAPDA -
#ClassAAPDA - #ClassAPDA _
                                ELSE BBAL("A1A", "A1B", "A2", "A3",
"A4", "M1", "M2", "M3") - #ClassAAAPDA - #ClassAAPDA - #ClassAPDA _
                                - MIN(COLL_BAL - #FloorOCTotal,
#AMTargPct * COLL_BAL)
    calculate : #ClassAMPDA = MAX( 0.0, MIN(BBAL("M3"), #ClassAMPDA ))
    calculate : #ClassAMPDA = MAX( 0, MIN( #ClassAMPDA, #DistribAmt -
#ClassAAAPDA - #ClassAAPDA - #ClassAPDA ) )
!
!
    calculate : #ClassBBBPPDA = IF (#TrigEvent OR (#StepDown EQ 0.0)) _
                                THEN #DistribAmt - #ClassAAAPDA -
#ClassAAPDA - #ClassAPDA - #ClassAMPDA _
                                ELSE BBAL("A1A", "A1B", "A2", "A3",
"A4", "M1", "M2", "M3", "M4") - #ClassAAAPDA - #ClassAAPDA - #ClassAPDA -
#ClassAMPDA _
                                - MIN(COLL_BAL - #FloorOCTotal,
#BBBPTargPct * COLL_BAL)
    calculate : #ClassBBBPPDA = MAX( 0.0, MIN(BBAL("M4"), #ClassBBBPPDA
))
    calculate : #ClassBBBPPDA = MAX( 0, MIN( #ClassBBBPPDA, #DistribAmt -
#ClassAAAPDA - #ClassAAPDA - #ClassAPDA - #ClassAMPDA ) )
!
!
    calculate : #ClassBBBPPDA = IF (#TrigEvent OR (#StepDown EQ 0.0)) _
                                THEN #DistribAmt - #ClassAAAPDA -
#ClassAAPDA - #ClassAPDA - #ClassAMPDA - #ClassBBBPPDA _
                                ELSE BBAL("A1A", "A1B", "A2", "A3",
"A4", "M1", "M2", "M3", "M4", "M5") - #ClassAAAPDA - #ClassAAPDA - #ClassAPDA -
#ClassAMPDA - #ClassBBBPPDA _
                                - MIN(COLL_BAL - #FloorOCTotal,
#BBBTargPct * COLL_BAL)
    calculate : #ClassBBBPPDA = MAX( 0.0, MIN(BBAL("M5"), #ClassBBBPPDA ))
    calculate : #ClassBBBPPDA = MAX( 0, MIN( #ClassBBBPPDA, #DistribAmt -
#ClassAAAPDA - #ClassAAPDA - #ClassAPDA - #ClassAMPDA - #ClassBBBPPDA ) )
!
!
    calculate : #ClassBBBMPDA = IF (#TrigEvent OR (#StepDown EQ 0.0)) _
                                THEN #DistribAmt - #ClassAAAPDA -
#ClassAAPDA - #ClassAPDA - #ClassAMPDA - #ClassBBBPPDA - #ClassBBBPPDA _
                                ELSE BBAL("A1A", "A1B", "A2", "A3",
"A4", "M1", "M2", "M3", "M4", "M5", "M6") - #ClassAAAPDA - #ClassAAPDA -
#ClassAPDA - #ClassAMPDA - #ClassBBBPPDA - #ClassBBBPPDA _
                                - MIN(COLL_BAL - #FloorOCTotal,
#BBBMTargPct * COLL_BAL)
    calculate : #ClassBBBMPDA = MAX( 0.0, MIN(BBAL("M6"), #ClassBBBMPDA
))
    calculate : #ClassBBBMPDA = MAX( 0, MIN( #ClassBBBMPDA, #DistribAmt -
#ClassAAAPDA - #ClassAAPDA - #ClassAPDA - #ClassAMPDA - #ClassBBBPPDA -
#ClassBBBPPDA ) )
!
!
    calculate : #ClassBBPPDA = IF (#TrigEvent OR (#StepDown EQ 0.0)) _

```

```

                                THEN  #DistribAmt  -  #ClassAAAPDA  -
#ClassAAPDA - #ClassAPDA - #ClassAMPDA - #ClassBBBPPDA - #ClassBBBPDA -
#ClassBBBMPDA _
                                ELSE  BBAL("A1A", "A1B", "A2", "A3",
"A4", "M1", "M2", "M3", "M4", "M5", "M6", "M7") - #ClassAAAPDA - #ClassAAPDA -
#ClassAPDA - #ClassAMPDA - #ClassBBBPPDA - #ClassBBBPDA - #ClassBBBMPDA _
                                - MIN(COLL_BAL - #FloorOCTotal,
#BBPTargPct * COLL_BAL)
    calculate : #ClassBBPPDA          = MAX( 0.0, MIN(BBAL("M7"), #ClassBBPPDA ))
    calculate : #ClassBBPPDA          = MAX( 0, MIN( #ClassBBPPDA, #DistribAmt -
#ClassAAAPDA - #ClassAAPDA - #ClassAPDA - #ClassAMPDA - #ClassBBBPPDA -
#ClassBBBPDA - #ClassBBBMPDA ) )
!
!
    calculate : "AAA" _
NO_CHECK CUSTOM AMOUNT          = #ClassAAAPDA
!
    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
!
    calculate : "BB+" _
NO_CHECK CUSTOM AMOUNT          = #ClassBBPPDA
!
-----
    from : CREDIT_ENHANCEMENT ("YmRsvFnd")
    subject to : CEILING ( (COLL("PREPAYPENALTY")) )
    pay : PREPAYPENALTY SEQUENTIAL("R_PP#1")
-----
!
----- Custom Payrule: Define CapCash1
calculate : #CapCash1 = #A1A_YM_INCOME
-----
!
----- Custom Payrule: Define CapCash2
calculate : #CapCash2 = #A1B_YM_INCOME
-----
!
----- Custom Payrule: Define CapCash3
calculate : #CapCash3 = #ANC_YM_INCOME
-----
!
----- Custom Payrule: Define CapCash4

```

```

calculate : #CapCash4      =      #SUB_YM_INCOME
-----
!
----- Pay Interest to AAA From Cash Account
      from : CLASS ( "ROOT" )
      pay : CLASS INTEREST SEQUENTIAL ( "AAA" )
-----
      {#AAA_Int}
-----
!
----- Pay Interest Shortfall to AAA From Cash
Account
      from : CLASS ( "ROOT" )
      pay : CLASS INTSHORT SEQUENTIAL ( "AAA" )
-----
      {#AAA_InS}
-----
!
----- Pay Interest to AA From Cash Account
      from : CLASS ( "ROOT" )
      pay : CLASS INTEREST SEQUENTIAL ( "AA" )
-----
!
----- Pay Interest Shortfall to AA From Cash
Account
      from : CLASS ( "ROOT" )
      pay : CLASS INTSHORT SEQUENTIAL ( "AA" )
-----
!
----- Pay Interest to A From Cash Account
      from : CLASS ( "ROOT" )
      pay : CLASS INTEREST SEQUENTIAL ( "A" )
-----
!
----- Pay Interest Shortfall to A From Cash
Account
      from : CLASS ( "ROOT" )
      pay : CLASS INTSHORT SEQUENTIAL ( "A" )
-----
!
----- Pay Interest to A- From Cash Account
      from : CLASS ( "ROOT" )
      pay : CLASS INTEREST SEQUENTIAL ( "A-" )
-----
!
----- Pay Interest Shortfall to A- From Cash
Account
      from : CLASS ( "ROOT" )
      pay : CLASS INTSHORT SEQUENTIAL ( "A-" )
-----
!
----- Pay Interest to BBB+ From Cash Account
      from : CLASS ( "ROOT" )
      pay : CLASS INTEREST SEQUENTIAL ( "BBB+" )
-----
!

```

```

----- Pay Interest Shortfall to BBB+ From Cash
Account
    from : CLASS ( "ROOT" )
    pay : CLASS INTSHORT SEQUENTIAL ( "BBB+" )
-----
!
----- Pay Interest to BBB From Cash Account
    from : CLASS ( "ROOT" )
    pay : CLASS INTEREST SEQUENTIAL ( "BBB" )
-----
!
----- Pay Interest Shortfall to BBB From Cash
Account
    from : CLASS ( "ROOT" )
    pay : CLASS INTSHORT SEQUENTIAL ( "BBB" )
-----
!
----- Pay Interest to BBB- From Cash Account
    from : CLASS ( "ROOT" )
    pay : CLASS INTEREST SEQUENTIAL ( "BBB-" )
-----
!
----- Pay Interest Shortfall to BBB- From Cash
Account
    from : CLASS ( "ROOT" )
    pay : CLASS INTSHORT SEQUENTIAL ( "BBB-" )
-----
!
----- Pay Interest to BB+ From Cash Account
    from : CLASS ( "ROOT" )
    pay : CLASS INTEREST SEQUENTIAL ( "BB+" )
-----
!
----- Pay Interest Shortfall to BB+ From Cash
Account
    from : CLASS ( "ROOT" )
    pay : CLASS INTSHORT SEQUENTIAL ( "BB+" )
-----
!
----- Pay Principal to AAA From Cash Account
    from : CLASS ( "ROOT" )
    pay : CLASS PRINCIPAL SEQUENTIAL ( "AAA" )
-----
{#AAA_Prn}{1}
-----
!
----- Pay Principal to AA From Cash Account
    from : CLASS ( "ROOT" )
    pay : CLASS PRINCIPAL SEQUENTIAL ( "AA" )
-----
{#AA_Prn}
-----
!
----- Pay Principal to A From Cash Account
    from : CLASS ( "ROOT" )
    pay : CLASS PRINCIPAL SEQUENTIAL ( "A" )
-----

```

```

{#A_Prn}
-----
!
----- Pay Principal to A- From Cash Account
      from : CLASS ( "ROOT" )
      pay  : CLASS PRINCIPAL SEQUENTIAL ( "A-" )
-----
{#AM_Prn}
-----
!
----- Pay Principal to BBB+ From Cash Account
      from : CLASS ( "ROOT" )
      pay  : CLASS PRINCIPAL SEQUENTIAL ( "BBB+" )
-----
{#BBBP_Prn}
-----
!
----- Pay Principal to BBB From Cash Account
      from : CLASS ( "ROOT" )
      pay  : CLASS PRINCIPAL SEQUENTIAL ( "BBB" )
-----
{#BBB_Prn}
-----
!
----- Pay Principal to BBB- From Cash Account
      from : CLASS ( "ROOT" )
      pay  : CLASS PRINCIPAL SEQUENTIAL ( "BBB-" )
-----
{#BBBM_Prn}
-----
!
----- Pay Principal to BB+ From Cash Account
      from : CLASS ( "ROOT" )
      pay  : CLASS PRINCIPAL SEQUENTIAL ( "BB+" )
-----
{#BBP_Prn}
-----
!
----- Pay Writedown Loss to AA From Cash
Account
      from : CLASS ( "ROOT" )
      pay  : CLASS PRINCSHORT_LOSS SEQUENTIAL ( "AA" )
-----
!
----- Pay Writedown Loss to A From Cash Account
      from : CLASS ( "ROOT" )
      pay  : CLASS PRINCSHORT_LOSS SEQUENTIAL ( "A" )
-----
!
----- Pay Writedown Loss to A- From Cash
Account
      from : CLASS ( "ROOT" )
      pay  : CLASS PRINCSHORT_LOSS SEQUENTIAL ( "A-" )
-----
!
----- Pay Writedown Loss to BBB+ From Cash
Account

```

```

        from : CLASS ( "ROOT" )
        pay : CLASS PRINCESHORT_LOSS SEQUENTIAL ( "BBB+" )
-----
!
----- Pay Writedown Loss to BBB From Cash
Account
        from : CLASS ( "ROOT" )
        pay : CLASS PRINCESHORT_LOSS SEQUENTIAL ( "BBB" )
-----
!
----- Pay Writedown Loss to BBB- From Cash
Account
        from : CLASS ( "ROOT" )
        pay : CLASS PRINCESHORT_LOSS SEQUENTIAL ( "BBB-" )
-----
!
----- Pay Writedown Loss to BB+ From Cash
Account
        from : CLASS ( "ROOT" )
        pay : CLASS PRINCESHORT_LOSS SEQUENTIAL ( "BB+" )
-----
!
----- Payback Basis Risk Shortfall From Cash
Account
        from : CLASS ( "ROOT" )
        pay : CLASS COUPONCAP_SHORT PRO_RATA ( "A1A"; "A1B"; "A2"; "A3"; "A4";
"AA"; "A"; "A-"; "BBB+"; "BBB"; "BBB-"; "BB+" )
-----
        from : HEDGE ("CAP")
        subject to : CEILING ((HEDGE ("CAP", "OPTIMAL_PMT") - HEDGE ("CAP",
"ACTUAL_PMT")))
        pay : CLASS COUPONCAP_SHORT PRO_RATA("A1A")
-----
        from : HEDGE ("CAP2")
        subject to : CEILING ((HEDGE ("CAP2", "OPTIMAL_PMT") - HEDGE ("CAP2",
"ACTUAL_PMT")))
        pay : CLASS COUPONCAP_SHORT PRO_RATA("A1B")
-----
        calculate : #A2_BAL_CCS = BBAL("A2") *
(COUPONCAP_ACCUM_SHORTFALL("A2") GT .01)
        calculate : #A3_BAL_CCS = BBAL("A3") *
(COUPONCAP_ACCUM_SHORTFALL("A3") GT .01)
        calculate : #A4_BAL_CCS = BBAL("A4") *
(COUPONCAP_ACCUM_SHORTFALL("A4") GT .01)
-----
        from : HEDGE ("CAP3")
        subject to : CEILING ((HEDGE ("CAP3", "OPTIMAL_PMT") - HEDGE ("CAP3",
"ACTUAL_PMT")))
        subject to : PROPORTION ( (#A2_BAL_CCS); (#A3_BAL_CCS); (#A4_BAL_CCS) )
        pay : CLASS COUPONCAP_SHORT ASIS CONCURRENT("A2" ; "A3" ; "A4")
-----
        from : HEDGE ("CAP3")
        subject to : CEILING ((HEDGE ("CAP3", "OPTIMAL_PMT") - HEDGE ("CAP3",
"ACTUAL_PMT")))
        pay : CLASS COUPONCAP_SHORT PRO_RATA("A2" ; "A3" ; "A4")
-----

```

```

calculate      :      #AA_BAL_CCS                                =      BBAL("AA")      *
(COUPONCAP_ACCUM_SHORTFALL("AA") GT .01)
calculate      :      #A_BAL_CCS                                =      BBAL("A")      *
(COUPONCAP_ACCUM_SHORTFALL("A") GT .01)
calculate      :      #AM_BAL_CCS                                =      BBAL("A-")      *
(COUPONCAP_ACCUM_SHORTFALL("A-") GT .01)
calculate      :      #BBBP_BAL_CCS                                =      BBAL("BBB+")      *
(COUPONCAP_ACCUM_SHORTFALL("BBB+") GT .01)
calculate      :      #BBB_BAL_CCS                                =      BBAL("BBB")      *
(COUPONCAP_ACCUM_SHORTFALL("BBB") GT .01)
calculate      :      #BBBM_BAL_CCS                                =      BBAL("BBB-")      *
(COUPONCAP_ACCUM_SHORTFALL("BBB-") GT .01)
calculate      :      #BBP_BAL_CCS                                =      BBAL("BB+")      *
(COUPONCAP_ACCUM_SHORTFALL("BB+") GT .01)

```

```

-----
from : HEDGE ("CAP4")
subject to : CEILING ((HEDGE ("CAP4", "OPTIMAL_PMT") - HEDGE ("CAP4",
"ACTUAL_PMT")))
subject to : PROPORTION ( (#AA_BAL_CCS); (#A_BAL_CCS); (#AM_BAL_CCS);
(#BBBP_BAL_CCS); (#BBB_BAL_CCS); (#BBBM_BAL_CCS); (#BBP_BAL_CCS) )
pay : CLASS COUPONCAP_SHORT ASIS CONCURRENT("AA" ; "A" ; "A-" ; "BBB+"
; "BBB" ; "BBB-" ; "BB+")

```

```

-----
from : HEDGE ("CAP4")
subject to : CEILING ((HEDGE ("CAP4", "OPTIMAL_PMT") - HEDGE ("CAP4",
"ACTUAL_PMT")))
pay : CLASS COUPONCAP_SHORT PRO_RATA("AA" ; "A" ; "A-" ; "BBB+" ;
"BBB" ; "BBB-" ; "BB+")

```

```

-----
from : HEDGE ("CAP")
pay : AS_INTEREST ("R#1")

```

```

-----
from : HEDGE ("CAP2")
pay : AS_INTEREST ("R#1")

```

```

-----
from : HEDGE ("CAP3")
pay : AS_INTEREST ("R#1")

```

```

-----
from : HEDGE ("CAP4")
pay : AS_INTEREST ("R#1")

```

!

```

from : CLASS ( "ROOT" )
pay : AS_INTEREST ("R#1")

```

```

-----
calculate      :      #WriteDown                                =      MAX(0.0,
BBAL("A1A#1", "A1B#1", "A2#1", "A3#1", "A4#1", "M1#1", "M2#1", "M3#1", "M4#1", "M5#1", "M6
#1", "M7#1") - COLL_BAL)

```

```

-----
from : SUBACCOUNT ( #Writedown )
pay : WRITEDOWN SEQUENTIAL ( "M7#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("A1A#1", "A1B#1", "A2#1", "A3#1", "A4#1", "M1#1", "M2#1", "M3#1", "M4#1", "M5#1", "M6
#1", "M7#1")
      calculate : #BondBal1      = BBAL("AAA_1")
      calculate : #BondBal2      = BBAL("AAA_2")
      calculate : #OC      = MAX( 0, COLL_BAL - #BondBal )
-----
!
!
      DEFINE DYNAMIC #YM_F11 = LOAN("SCHAM_PREP_AMT") * 80% * LOAN("GROSSRATE") /
(36500/30) * 2
!
      DEFINE DYNAMIC #YM_F16 = MIN(2% * LOAN("SCHAM_PREP_AMT"),
LOAN("SCHAM_PREP_AMT") * LOAN("GROSSRATE") / 1200 * 2)
!
      DEFINE DYNAMIC #YM_F2 = LOAN("SCHAM_PREP_AMT") * 100% * LOAN("GROSSRATE") /
1200 * 3
!
      DEFINE DYNAMIC #YM_F22 = if (curmonth <= 12) then LOAN("SCHAM_PREP_AMT")*3%
else if (curmonth <= 24) then LOAN("SCHAM_PREP_AMT")*2% else
LOAN("SCHAM_PREP_AMT")*1%
!
      DEFINE DYNAMIC #YM_F21 = 1%*80%*LOAN("SCHAM_PREP_AMT")
!
      DEFINE DYNAMIC #YM_F4 = if (curmonth <= 12) then LOAN("SCHAM_PREP_AMT")*5%
else if (curmonth <= 24) then LOAN("SCHAM_PREP_AMT")*4% else
LOAN("SCHAM_PREP_AMT")*3%
!
      DEFINE DYNAMIC #YM_F5 = 2%*LOAN("SCHAM_PREP_AMT")
!
      DEFINE DYNAMIC #YM_HE6MO80PCT = LOAN("SCHAM_PREP_AMT") * 80% *
LOAN("GROSSRATE") / 1200 * 6
!
      DEFINE DYNAMIC #YM_F8 = 1%*LOAN("SCHAM_PREP_AMT")
!
      DEFINE DYNAMIC #YM_F15 = if (curmonth <= 12) then LOAN("SCHAM_PREP_AMT")*2%
else if (curmonth <= 24) then LOAN("SCHAM_PREP_AMT")*1% else 0
!
      DEFINE DYNAMIC #YM_A22 = if (curmonth <= 12) then LOAN("SCHAM_PREP_AMT")*3%
else if (curmonth <= 24) then LOAN("SCHAM_PREP_AMT")*2% else
LOAN("SCHAM_PREP_AMT")*1%
!
      DEFINE DYNAMIC #YM_A21 = 1%*80%*LOAN("SCHAM_PREP_AMT")
!

```

```

DEFINE DYNAMIC #YM_A5 = 2%* LOAN("SCHAM_PREP_AMT")
!
Collateral OVER
!
!      Factor      --Delay--
! Type   Date      P/Y    BV    Use BV for 0
! WL 20040601      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      "Conforming_FRM_11"      WL      00      WAC
6.982 (      4899417.84 /      4899417.84 );      4899417.84
0.5022      0.5022      322:1      322:1      323 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F11 );
M      2      "Conforming_FRM_13"      WL      00      WAC
7.002 (      5170137.56 /      5170137.56 );      5170137.56
0.5022      0.5022      328:2      328:2      330 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F16 );
M      3      "Conforming_FRM_2"      WL      00      WAC
6.903 (      4966925.11 /      4966925.11 );      4966925.11
0.5022      0.5022      349:1      349:1      350 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F2 );
M      4      "Conforming_FRM_22"      WL      00      WAC
8.008 (      666085.61 /      666085.61 );      666085.61
0.5022      0.5022      359:1      359:1      360 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F22 );
M      5      "Conforming_FRM_3,21"      WL      00      WAC
6.874 (      633422.57 /      633422.57 );      633422.57
0.5022      0.5022      359:1      359:1      360 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA ( #YM_F21 );
M      6      "Conforming_FRM_"      WL      00      WAC
7.01 (      3830925.50 /      3830925.50 );      3830925.50
0.5022      0.5022      352:1      352:1      353 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F21 );
M      7      "Conforming_FRM_4"      WL      00      WAC
7.755 (      2287250.35 /      2287250.35 );      2287250.35
0.5022      0.5022      318:1      318:1      319 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F4 );
M      8      "Conforming_FRM_5,10"      WL      00      WAC
6.774 (      2728053.59 /      2728053.59 );      2728053.59
0.5022      0.5022      337:1      337:1      338 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA ( #YM_F5 );
M      9      "Conforming_FRM_"      WL      00      WAC
7.001 (      2777631.16 /      2777631.16 );      2777631.16
0.5022      0.5022      335:1      335:1      336 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F5 );
M      10      "Conforming_FRM_6,7,9,STD"      WL      00      WAC
6.879 (      17294687.36 /      17294687.36 );      17294687.36
0.5022      0.5022      334:1      334:1      335 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA ( #YM_HE6MO80PCT );
M      11      "Conforming_FRM_"      WL      00      WAC
6.854 (      90080229.69 /      90080229.69 );      90080229.69
0.5022      0.5022      339:1      339:1      340 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_HE6MO80PCT );

```

M 12 "Conforming_FRM_8" WL 00 WAC
7.243 (2289205.37 / 2289205.37); 2289205.37
0.5022 0.5022 347:1 347:1 348 NO_CHECK
GROUP "1F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_F8);
M 13 "Conforming_FRM_No Penalty" WL 00 WAC
7.663 (29927238.73 / 29927238.73); 29927238.73
0.5022 0.5022 321:1 321:1 322 NO_CHECK
GROUP "1F" PREPAY_FLAG NONE ;
M 14 "Conforming_ARM_13" WL 00 WAC
7.389 (22485362.02 / 22485362.02); 22485362.02
0.5022 0.5022 356:1 356:1 357 NO_CHECK ARM LIBOR_6MO
6.007 24 6 SYNC_INT 13.389 1 7.389
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 7.389 GROUP "1A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_F16); TEASER
M 15 "Conforming_ARM_15" WL 00 WAC
8.981 (587952.18 / 587952.18); 587952.18
0.5022 0.5022 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.94 24 6 SYNC_INT 14.981 1 8.981
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 8.981 GROUP "1A"
PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_F15); TEASER
M 16 "Conforming_ARM_2" WL 00 WAC
7.227 (43450736.47 / 43450736.47); 43450736.47
0.5022 0.5022 357:1 357:1 358 NO_CHECK ARM LIBOR_6MO
6.021 24 6 SYNC_INT 13.227 1 7.227
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 7.227 GROUP "1A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_F2); TEASER
M 17 "Conforming_ARM_22" WL 00 WAC
8.275 (1387128.60 / 1387128.60); 1387128.60
0.5022 0.5022 349:3 349:3 352 NO_CHECK ARM LIBOR_6MO
5.802 22 6 SYNC_INT 14.245 1 8.245
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 8.275 GROUP "1A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_A22); TEASER
M 18 "Conforming_ARM_3,21" WL 00 WAC
8.383 (1882863.27 / 1882863.27); 1882863.27
0.5022 0.5022 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
6.301 24 6 SYNC_INT 14.383 1 8.383
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 8.383 GROUP "1A"
PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (#YM_A21); TEASER
M 19 "Conforming_ARM_" WL 00 WAC
7.5 (30346978.76 / 30346978.76); 30346978.76
0.5022 0.5022 358:1 358:1 359 NO_CHECK ARM LIBOR_6MO
5.99 24 6 SYNC_INT 13.5 1 7.5
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 7.5 GROUP "1A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_A21); TEASER
M 20 "Conforming_ARM_4" WL 00 WAC
8.015 (6639271.71 / 6639271.71); 6639271.71
0.5022 0.5022 349:1 349:1 350 NO_CHECK ARM LIBOR_6MO
6.139 24 6 SYNC_INT 14.015 1 8.015
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 8.015 GROUP "1A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_F4); TEASER
M 21 "Conforming_ARM_5,10" WL 00 WAC
7.05 (11551424.01 / 11551424.01); 11551424.01
0.5022 0.5022 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.965 24 6 SYNC_INT 13.05 1 7.05
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 7.05 GROUP "1A"
PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_A5); TEASER

M 22 "Conforming_ARM_" WL 00 WAC
7.819 (8043946.26 / 8043946.26); 8043946.26
0.5022 0.5022 355:1 355:1 356 NO_CHECK ARM LIBOR_6MO
6.135 24 6 SYNC_INT 13.819 1 7.819
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 7.819 GROUP "1A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_A5); TEASER
M 23 "Conforming_ARM_6,7,9,STD" WL 00 WAC
6.99 (188688.63 / 188688.63); 188688.63
0.5022 0.5022 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
6 23 6 SYNC_INT 12.99 1 6.99
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 6.99 GROUP "1A"
PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 24 "Conforming_ARM_" WL 00 WAC
7.869 (668220.79 / 668220.79); 668220.79
0.5022 0.5022 357:3 357:3 360 NO_CHECK ARM LIBOR_6MO
5.75 22 6 SYNC_INT 13.869 1 7.869
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 7.869 GROUP "1A"
PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 25 "Conforming_ARM_" WL 00 WAC
7.254 (310440914.98 / 310440914.98); 310440914.98
0.5022 0.5022 355:2 355:2 357 NO_CHECK ARM LIBOR_6MO
6.049 24 6 SYNC_INT 13.254 1 7.254
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 7.254 GROUP "1A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 26 "Conforming_ARM_8" WL 00 WAC
7.725 (13937594.15 / 13937594.15); 13937594.15
0.5022 0.5022 356:1 356:1 357 NO_CHECK ARM LIBOR_6MO
6.13 24 6 SYNC_INT 13.725 1 7.725
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 7.725 GROUP "1A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_F8); TEASER
M 27 "Conforming_ARM_No Penalty" WL 00 WAC
7.95 (208482133.09 / 208482133.09); 208482133.09
0.5022 0.5022 355:1 355:1 356 NO_CHECK ARM LIBOR_6MO
6.034 24 6 SYNC_INT 13.95 1 7.95
0 0 INIT_PERCAP 2 ORIG_GROSSRATE 7.95 GROUP "1A"
PREPAY_FLAG NONE ; TEASER
M 28 "Non-Conforming_FRM_11" WL 00 WAC
8.201 (188228.66 / 188228.66); 188228.66
0.5022 0.5022 359:1 359:1 360 NO_CHECK
GROUP "2F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_F11);
M 29 "Non-Conforming_FRM_13" WL 00 WAC
7.5 (503625.95 / 503625.95); 503625.95
0.5022 0.5022 359:1 359:1 360 NO_CHECK
GROUP "2F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_F16);
M 30 "Non-Conforming_FRM_2" WL 00 WAC
6.605 (1951973.20 / 1951973.20); 1951973.20
0.5022 0.5022 352:1 352:1 353 NO_CHECK
GROUP "2F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_F2);
M 31 "Non-Conforming_FRM_22" WL 00 WAC
7.85 (76447.08 / 76447.08); 76447.08
0.5022 0.5022 359:1 359:1 360 NO_CHECK
GROUP "2F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_F22);
M 32 "Non-Conforming_FRM_3,21" WL 00 WAC
6.4 (352674.63 / 352674.63); 352674.63
0.5022 0.5022 359:1 359:1 360 NO_CHECK
GROUP "2F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_F21);

```

M          33      "Non-Conforming_FRM_4"                      WL      00      WAC
6.786 (          399951.61 /          399951.61 );          399951.61
0.5022      0.5022          319:1          319:1          320 NO_CHECK
GROUP "2F"    PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F4 );
M          34      "Non-Conforming_FRM_5,10"                  WL      00      WAC
8.046 (          307493.67 /          307493.67 );          307493.67
0.5022      0.5022          359:1          359:1          360 NO_CHECK
GROUP "2F"    PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA ( #YM_F5 );
M          35      "Non-Conforming_FRM_6,7,9,STD"              WL      00      WAC
6.909 (          6050890.42 /          6050890.42 );          6050890.42
0.5022      0.5022          348:1          348:1          349 NO_CHECK
GROUP "2F"    PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA ( #YM_HE6MO80PCT );
M          36      "Non-Conforming_FRM_"                      WL      00      WAC
6.577 (          17990327.05 /          17990327.05 );          17990327.05
0.5022      0.5022          351:1          351:1          352 NO_CHECK
GROUP "2F"    PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_HE6MO80PCT );
M          37      "Non-Conforming_FRM_8"                      WL      00      WAC
6.985 (          874458.00 /          874458.00 );          874458.00
0.5022      0.5022          291:1          291:1          292 NO_CHECK
GROUP "2F"    PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F8 );
M          38      "Non-Conforming_FRM_No Penalty"              WL      00      WAC
7.199 (          3806042.65 /          3806042.65 );          3806042.65
0.5022      0.5022          345:1          345:1          346 NO_CHECK
GROUP "2F"    PREPAY_FLAG NONE ;
M          39      "Non-Conforming_ARM_11"                      WL      00      WAC
10.5 (          59853.73 /          59853.73 );          59853.73
0.5022      0.5022          354:6          354:6          360 NO_CHECK ARM LIBOR_6MO
6      19      6 SYNC_INT          16.5          1          10.5
0      0      INIT_PERCAP          2      ORIG_GROSSRATE          10.5 GROUP "2A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F11 ); TEASER
M          40      "Non-Conforming_ARM_13"                      WL      00      WAC
7.523 (          2973576.03 /          2973576.03 );          2973576.03
0.5022      0.5022          359:1          359:1          360 NO_CHECK ARM LIBOR_6MO
6.182      24      6 SYNC_INT          13.523          1          7.523
0      0      INIT_PERCAP          2      ORIG_GROSSRATE          7.523 GROUP "2A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F16 ); TEASER
M          41      "Non-Conforming_ARM_15"                      WL      00      WAC
7.398 (          1680574.24 /          1680574.24 );          1680574.24
0.5022      0.5022          359:1          359:1          360 NO_CHECK ARM LIBOR_6MO
5.874      24      6 SYNC_INT          13.398          1          7.398
0      0      INIT_PERCAP          2      ORIG_GROSSRATE          7.398 GROUP "2A"
PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA ( #YM_F15 ); TEASER
M          42      "Non-Conforming_ARM_2"                      WL      00      WAC
7.106 (          6005446.16 /          6005446.16 );          6005446.16
0.5022      0.5022          359:1          359:1          360 NO_CHECK ARM LIBOR_6MO
5.99      24      6 SYNC_INT          13.106          1          7.106
0      0      INIT_PERCAP          2      ORIG_GROSSRATE          7.106 GROUP "2A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F2 ); TEASER
M          43      "Non-Conforming_ARM_22"                      WL      00      WAC
11.4 (          71976.47 /          71976.47 );          71976.47
0.5022      0.5022          359:1          359:1          360 NO_CHECK ARM LIBOR_6MO
6.5      24      6 SYNC_INT          17.4          1          11.4
0      0      INIT_PERCAP          2      ORIG_GROSSRATE          11.4 GROUP "2A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_A22 ); TEASER
M          44      "Non-Conforming_ARM_3,21"                  WL      00      WAC
9.8 (          157427.29 /          157427.29 );          157427.29
0.5022      0.5022          359:1          359:1          360 NO_CHECK ARM LIBOR_6MO

```

```

6.5      24      6 SYNC_INT              15.8              1              9.8
0        0 INIT_PERCAP                  2 ORIG_GROSSRATE          9.8 GROUP "2A"
PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA ( #YM_A21 ); TEASER
M        45      "Non-Conforming_ARM_"          WL      00      WAC
7.588 (      2695616.37 /      2695616.37 );      2695616.37
0.5022      0.5022      359:1      359:1      360 NO_CHECK ARM LIBOR_6MO
6.071      24      6 SYNC_INT              13.588              1              7.588
0        0 INIT_PERCAP                  2 ORIG_GROSSRATE          7.588 GROUP "2A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_A21 ); TEASER
M        46      "Non-Conforming_ARM_4"          WL      00      WAC
8.361 (      891829.35 /      891829.35 );      891829.35
0.5022      0.5022      330:1      330:1      331 NO_CHECK ARM LIBOR_6MO
6.192      24      6 SYNC_INT              14.361              1              8.361
0        0 INIT_PERCAP                  2 ORIG_GROSSRATE          8.361 GROUP "2A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F4 ); TEASER
M        47      "Non-Conforming_ARM_5,10"          WL      00      WAC
7.948 (      1345334.01 /      1345334.01 );      1345334.01
0.5022      0.5022      359:1      359:1      360 NO_CHECK ARM LIBOR_6MO
6.55      24      6 SYNC_INT              13.948              1              7.948
0        0 INIT_PERCAP                  2 ORIG_GROSSRATE          7.948 GROUP "2A"
PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA ( #YM_A5 ); TEASER
M        48      "Non-Conforming_ARM_"          WL      00      WAC
8.044 (      1113407.22 /      1113407.22 );      1113407.22
0.5022      0.5022      358:2      358:2      360 NO_CHECK ARM LIBOR_6MO
6.396      23      6 SYNC_INT              14.044              1              8.044
0        0 INIT_PERCAP                  2 ORIG_GROSSRATE          8.044 GROUP "2A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_A5 ); TEASER
M        49      "Non-Conforming_ARM_6,7,9,STD"          WL      00      WAC
6.917 (      75407369.54 /      75407369.54 );      75407369.54
0.5022      0.5022      358:1      358:1      359 NO_CHECK ARM LIBOR_6MO
5.907      24      6 SYNC_INT              12.917              1              6.917
0        0 INIT_PERCAP                  2 ORIG_GROSSRATE          6.917 GROUP "2A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_HE6MO80PCT ); TEASER
M        50      "Non-Conforming_ARM_8"          WL      00      WAC
7.851 (      1030028.73 /      1030028.73 );      1030028.73
0.5022      0.5022      359:1      359:1      360 NO_CHECK ARM LIBOR_6MO
6.238      24      6 SYNC_INT              13.851              1              7.851
0        0 INIT_PERCAP                  2 ORIG_GROSSRATE          7.851 GROUP "2A"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA ( #YM_F8 ); TEASER
M        51      "Non-Conforming_ARM_No Penalty"          WL      00      WAC
7.578 (      46421431.61 /      46421431.61 );      46421431.61
0.5022      0.5022      357:1      357:1      358 NO_CHECK ARM LIBOR_6MO
5.871      24      6 SYNC_INT              13.578              1              7.578
0        0 INIT_PERCAP                  2 ORIG_GROSSRATE          7.578 GROUP "2A"
PREPAY_FLAG NONE ; TEASER

```