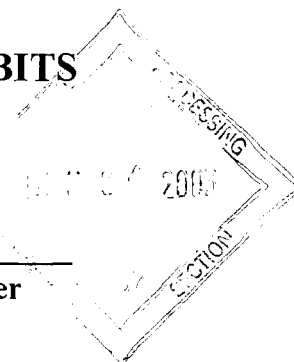


FORM SE
FORM FOR SUBMISSION OF PAPER FORMAT EXHIBITS
BY ELECTRONIC FILERS

Argent Securities Inc.
Exact Name of Registrant as Specified in Charter
Form 8-K, December 3, 2003, Series 2003-W8

0001239602
Registrant CIK Number
333-109164



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



PROCESSED
DEC 08 2003
THOMSON
FINANCIAL

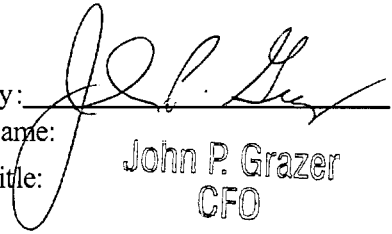
3

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: December 3, 2003

ARGENT 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>
99.3	Computational Materials	P*

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

! ARSI2003W8TS.CDI #CMOVER_3.0B ASSET_BACKED_HOMEEQUITY !
MAX_CF_VECTSIZE 553

!

!! Created by Intex Deal Maker v3.5.359 , subroutines 3.0f

!! 11/05/2003 4:05 PM

!

COLLAT_TYPE "Fixed and Adjustable Rate Mortgages"

!

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 Banc of America Securities LLC provided the computer model used to generate them). The Hypothetical Performance Data will be generated by you using a computer model prepared by Banc of America Securities 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 Banc of America Securities LLC or any other person. The computer model that you will use to prepare the Hypothetical Performance Data was furnished to you solely by Banc of America Securities 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 Banc of America Securities 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 or 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 Banc of America Securities LLC Trading Desk at (704) 388-1579 to confirm the final principal amount, designation and terms of any security described in this communication prior to


```

!
DEFINE TR_INDEXDEPS_ALL
!
    DEAL_CLOCK_INFO _
        ISSUE_CDU_DATE          20031201 _
        DEAL_FIRSTPAY_DATE      20040125 _
!
!
DEFINE TABLE "CapNotional" (1, 2) = "CURDATE" "Balance"
    20040125.1      347,490,000
!
DEFINE TABLE "CapRateSch" (1, 2) = "CURDATE" "Rate"
    20040125.1      8.89
!
    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" (40, 2) = "CURDATE" "Balance"
    20040125.1      38,610,000
    20040225.1      37,824,082
    20040325.1      36,790,287
    20040425.1      35,760,665
    20040525.1      34,734,621
    20040625.1      33,711,685
    20040725.1      32,691,512
    20040825.1      31,673,883
    20040925.1      30,658,892
    20041025.1      29,663,337
    20041125.1      28,686,833
    20041225.1      27,728,918
    20050125.1      26,794,822
    20050225.1      25,883,949
    20050325.1      24,995,716
    20050425.1      24,129,555
    20050525.1      23,284,913
    20050625.1      22,461,251
    20050725.1      21,658,043
    20050825.1      20,874,776
    20050925.1      20,110,952
    20051025.1      19,366,083
    20051125.1      18,639,694
    20051225.1      17,933,882
    20060125.1      17,245,946
    20060225.1      16,574,995
    20060325.1      15,921,554
    20060425.1      15,284,216
    20060525.1      14,662,579
    20060625.1      14,057,124
    20060725.1      13,466,706
    20060825.1      12,890,800
    20060925.1      12,329,369
    20061025.1      11,781,724
    20061125.1      11,247,523
    20061225.1      10,727,078

```

20070125.1	10,220,331
20070225.1	10,220,331
20070325.1	10,220,331
20070425.1	10,220,331

!

DEFINE TABLE "Cap2RateSch" (40, 2) = "CURDATE" "Rate"

20040125.1	8.75
20040225.1	6.12
20040325.1	6.58
20040425.1	6.12
20040525.1	6.34
20040625.1	6.12
20040725.1	6.34
20040825.1	6.12
20040925.1	6.12
20041025.1	6.34
20041125.1	6.12
20041225.1	6.34
20050125.1	6.12
20050225.1	6.11
20050325.1	6.82
20050425.1	6.11
20050525.1	6.33
20050625.1	6.11
20050725.1	6.33
20050825.1	6.11
20050925.1	6.11
20051025.1	6.33
20051125.1	6.75
20051225.1	7.10
20060125.1	6.85
20060225.1	7.11
20060325.1	7.92
20060425.1	7.10
20060525.1	7.68
20060625.1	7.46
20060725.1	7.73
20060825.1	7.58
20060925.1	7.58
20061025.1	7.84
20061125.1	7.89
20061225.1	8.48
20070125.1	8.19
20070225.1	8.40
20070325.1	8.75
20070425.1	8.39

!

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" (49, 2) = "CURDATE" "Balance"

20040125.1	207,900,000
20040225.1	204,222,522
20040325.1	199,366,420

20040425.1	194,442,154
20040525.1	189,447,531
20040625.1	184,381,341
20040725.1	179,243,373
20040825.1	174,035,863
20040925.1	168,783,036
20041025.1	163,607,307
20041125.1	158,506,842
20041225.1	153,479,995
20050125.1	148,573,207
20050225.1	143,783,566
20050325.1	139,108,236
20050425.1	134,544,446
20050525.1	130,089,496
20050625.1	125,740,752
20050725.1	121,495,643
20050825.1	117,351,664
20050925.1	113,306,369
20051025.1	109,357,373
20051125.1	105,502,349
20051225.1	101,750,941
20060125.1	98,090,458
20060225.1	94,516,638
20060325.1	91,031,820
20060425.1	87,629,334
20060525.1	84,307,205
20060625.1	81,067,560
20060725.1	77,904,965
20060825.1	74,816,868
20060925.1	71,802,994
20061025.1	68,860,027
20061125.1	65,986,275
20061225.1	63,183,094
20070125.1	60,449,168
20070225.1	60,449,168
20070325.1	60,449,168
20070425.1	60,449,168
20070525.1	60,449,168
20070625.1	59,755,240
20070725.1	58,373,258
20070825.1	57,023,519
20070925.1	55,705,938
20071025.1	54,419,059
20071125.1	53,162,158
20071225.1	51,935,504
20080125.1	50,737,992

!

DEFINE TABLE "Cap3RateSch" (49, 2) = "CURDATE" "Rate"

20040125.1	8.86
20040225.1	6.18
20040325.1	6.64
20040425.1	6.18
20040525.1	6.40
20040625.1	6.18
20040725.1	6.40
20040825.1	6.18
20040925.1	6.18
20041025.1	6.39
20041125.1	6.17
20041225.1	6.39
20050125.1	6.17

20050225.1	6.17
20050325.1	6.87
20050425.1	6.17
20050525.1	6.39
20050625.1	6.17
20050725.1	6.39
20050825.1	6.17
20050925.1	6.17
20051025.1	6.39
20051125.1	6.70
20051225.1	7.03
20060125.1	6.78
20060225.1	6.99
20060325.1	7.78
20060425.1	6.99
20060525.1	7.50
20060625.1	7.29
20060725.1	7.54
20060825.1	7.38
20060925.1	7.38
20061025.1	7.63
20061125.1	7.62
20061225.1	8.10
20070125.1	7.82
20070225.1	7.97
20070325.1	8.86
20070425.1	7.96
20070525.1	8.49
20070625.1	8.32
20070725.1	8.60
20070825.1	8.43
20070925.1	8.43
20071025.1	8.71
20071125.1	8.66
20071225.1	8.86
20080125.1	8.76

!

```

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" (96, 2) = "CURDATE" "Balance"
20040125.1      136,875,000
20040225.1      136,875,000
20040325.1      136,875,000
20040425.1      136,875,000
20040525.1      136,875,000
20040625.1      136,875,000
20040725.1      136,875,000
20040825.1      136,875,000
20040925.1      136,875,000
20041025.1      136,875,000
20041125.1      136,875,000
20041225.1      136,875,000
20050125.1      136,875,000
```

20050225.1	136,875,000
20050325.1	136,875,000
20050425.1	136,875,000
20050525.1	136,875,000
20050625.1	136,875,000
20050725.1	136,875,000
20050825.1	136,875,000
20050925.1	136,875,000
20051025.1	136,875,000
20051125.1	136,875,000
20051225.1	136,875,000
20060125.1	136,875,000
20060225.1	136,875,000
20060325.1	136,875,000
20060425.1	136,875,000
20060525.1	136,875,000
20060625.1	136,875,000
20060725.1	136,875,000
20060825.1	136,875,000
20060925.1	136,875,000
20061025.1	136,875,000
20061125.1	136,875,000
20061225.1	136,875,000
20070125.1	136,875,000
20070225.1	132,523,618
20070325.1	125,479,379
20070425.1	118,604,631
20070525.1	111,895,249
20070625.1	106,047,827
20070725.1	101,050,389
20070825.1	98,230,998
20070925.1	95,895,095
20071025.1	93,615,170
20071125.1	91,389,865
20071225.1	89,219,779
20080125.1	87,102,999
20080225.1	85,036,813
20080325.1	83,021,055
20080425.1	81,053,424
20080525.1	79,132,758
20080625.1	77,257,921
20080725.1	75,428,602
20080825.1	73,642,890
20080925.1	71,899,990
20081025.1	70,198,605
20081125.1	68,537,736
20081225.1	66,916,405
20090125.1	65,334,260
20090225.1	63,789,745
20090325.1	62,282,149
20090425.1	60,810,384
20090525.1	59,373,588
20090625.1	57,970,923
20090725.1	56,601,566
20090825.1	55,264,718
20090925.1	53,959,598
20091025.1	52,685,445
20091125.1	51,441,515
20091225.1	50,227,083
20100125.1	49,041,442
20100225.1	47,883,901

20100325.1	46,753,788
20100425.1	45,650,444
20100525.1	44,573,229
20100625.1	43,521,516
20100725.1	42,494,697
20100825.1	41,492,175
20100925.1	40,513,368
20101025.1	39,557,710
20101125.1	38,624,648
20101225.1	37,713,641
20110125.1	36,824,162
20110225.1	35,955,699
20110325.1	35,107,748
20110425.1	34,279,820
20110525.1	33,471,438
20110625.1	32,682,136
20110725.1	31,911,459
20110825.1	31,158,962
20110925.1	30,424,213
20111025.1	29,706,789
20111125.1	29,006,278
20111225.1	28,322,276

!

DEFINE TABLE "Cap4RateSch" (96, 2) = "CURDATE" "Rate"

20040125.1	9.18
20040225.1	4.79
20040325.1	5.24
20040425.1	4.78
20040525.1	5.00
20040625.1	4.78
20040725.1	5.00
20040825.1	4.78
20040925.1	4.78
20041025.1	5.00
20041125.1	4.78
20041225.1	5.00
20050125.1	4.78
20050225.1	4.78
20050325.1	5.48
20050425.1	4.78
20050525.1	4.99
20050625.1	4.77
20050725.1	4.99
20050825.1	4.77
20050925.1	4.77
20051025.1	4.99
20051125.1	5.38
20051225.1	5.72
20060125.1	5.47
20060225.1	5.71
20060325.1	6.51
20060425.1	5.70
20060525.1	6.26
20060625.1	6.04
20060725.1	6.30
20060825.1	6.15
20060925.1	6.15
20061025.1	6.41
20061125.1	6.44
20061225.1	6.99
20070125.1	6.70

20070225.1	6.89
20070325.1	7.82
20070425.1	6.88
20070525.1	7.46
20070625.1	7.31
20070725.1	7.61
20070825.1	7.45
20070925.1	7.44
20071025.1	7.74
20071125.1	7.71
20071225.1	8.18
20080125.1	7.85
20080225.1	7.99
20080325.1	8.66
20080425.1	7.97
20080525.1	8.29
20080625.1	8.06
20080725.1	8.38
20080825.1	8.08
20080925.1	8.07
20081025.1	8.39
20081125.1	8.05
20081225.1	8.48
20090125.1	8.14
20090225.1	8.17
20090325.1	9.18
20090425.1	8.15
20090525.1	8.47
20090625.1	8.13
20090725.1	8.45
20090825.1	8.11
20090925.1	8.10
20091025.1	8.42
20091125.1	8.08
20091225.1	8.40
20100125.1	8.06
20100225.1	8.05
20100325.1	9.09
20100425.1	8.03
20100525.1	8.35
20100625.1	8.01
20100725.1	8.33
20100825.1	7.99
20100925.1	7.98
20101025.1	8.30
20101125.1	7.96
20101225.1	8.28
20110125.1	7.94
20110225.1	7.93
20110325.1	8.96
20110425.1	7.91
20110525.1	8.23
20110625.1	7.89
20110725.1	8.21
20110825.1	7.87
20110925.1	7.86
20111025.1	8.18
20111125.1	7.84
20111225.1	8.16

!

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          = 0.5% * #OrigCollBal
  DEFINE #ReqPerc              = 0
  DEFINE #TrigEnhFrac          = 0
  DEFINE #CumLossShft         = 0
  DEFINE #TrigCumLossFrac      = 0
  DEFINE #SpecOCTarg          = 2.55% * #OrigCollBal
ifnndef #cmover_3.0d _
  DEFINE #OC                  = 19125000.00
!
ifdef #cmover_3.0d _
  DEFINE STANDARDIZE OC_ACTUAL_VAL          #OC          =
19125000.00
!
  DEFINE STANDARDIZE OCT_INITVAL          CONSTANT #InitOCTarg    = 2.55%
* #OrigCollBal
  DEFINE STANDARDIZE OCT_STEPDOWN_MONTH  CONSTANT #StepDownDate  = 37
  DEFINE STANDARDIZE OCT_STEPDOWN_FRAC   CONSTANT #StepOCFrac    = 0.051
  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 #NetRateActual360 = #Netrate * 30 / DAYS_DIFF
(CURDATE , MONTHS_ADD(CURDATE,-1))
!
!
  DEFINE TABLE "OC_CUMLOSS0" (5, 2) = "MONTH" "OC_CUMLOSS_FRAC0"
    48.1    0.0325
    60.1    0.0525
    72.1    0.0675
    84.1    0.075
    360.1   0.0775
!
!
TOLERANCE WRITEDOWN_OLOSS 1.00
!
  INITIAL INDEX    LIBOR_6MO          1.24
  INITIAL INDEX    LIBOR_1MO          1.12
!
  DEFINE TRANCHE "CAP_IN", "CAP_IN2", "CAP_IN3", "CAP_IN4", "A1", "A2",
"A3", "M1", "M2", "M3", "M4", "M5", "M6", "OC", "P"
!
!
Tranche "CAP_IN" PSEUDO HEDGE

```

```

Block $ 347,490,000 at 0.00 FLOAT NOTIONAL WITH FORMULA BEGIN (
#CapBal ); END ( #CapBalEnd );
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M
Delay 0 Dated 20031204 Next 20040125
((1 * MIN(8.89, LIBOR_1MO)) + (-1 * #CapRate))
0 999
!
Tranche "CAP_IN2" PSEUDO HEDGE
Block $ 38,610,000 at 0.00 FLOAT NOTIONAL WITH FORMULA BEGIN (
#CapBal2 ); END ( #CapBal2End );
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M
Delay 0 Dated 20031204 Next 20040125
((1 * MIN(8.75, LIBOR_1MO)) + (-1 * #CapRate2))
0 999
!
Tranche "CAP_IN3" PSEUDO HEDGE
Block $ 207,900,000 at 0.00 FLOAT NOTIONAL WITH FORMULA BEGIN (
#CapBal3 ); END ( #CapBal3End );
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M
Delay 0 Dated 20031204 Next 20040125
((1 * MIN(8.86, LIBOR_1MO)) + (-1 * #CapRate3))
0 999
!
Tranche "CAP_IN4" PSEUDO HEDGE
Block $ 136,875,000 at 0.00 FLOAT NOTIONAL WITH FORMULA BEGIN (
#CapBal4 ); END ( #CapBal4End );
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M
Delay 0 Dated 20031204 Next 20040125
((1 * MIN(9.18, LIBOR_1MO)) + (-1 * #CapRate4))
0 999
!
Tranche "A1" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 347490000.00 at 1.48 GROUP 1 FREQ M FLOAT RESET M
COUPONCAP 30360 NONE ( 1200* COLL_I_MISC
("COUPON",1)/COLL_PREV_BAL(1) );
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
Delay 0 Dated 20031204 Next 20040125
(1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 0.72 ELSE 0.36 ))
0 999
!
Tranche "A2" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 38610000.00 at 1.62 GROUP 1 FREQ M FLOAT RESET M
COUPONCAP 30360 NONE ( 1200* COLL_I_MISC
("COUPON",1)/COLL_PREV_BAL(1) );
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
Delay 0 Dated 20031204 Next 20040125
(1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 1. ELSE 0.5 ))
0 999
!
Tranche "A3" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
Block 207900000.00 at 1.51 GROUP 2 FREQ M FLOAT RESET M
COUPONCAP 30360 NONE ( 1200* COLL_I_MISC
("COUPON",2)/COLL_PREV_BAL(2) );
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
Delay 0 Dated 20031204 Next 20040125
(1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 0.78 ELSE 0.39 ))
0 999
!

```

```

Tranche "M1" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 50625000.00 at 1.81  FREQ M FLOAT RESET M _
    COUPONCAP 30360 NONE ( 1200*(COLL_I_MISC
("COUPON",1)/COLL_PREV_BAL(1)*(COLL_PREV_BAL(1)-BBAL("A1","A2"))
+COLL_I_MISC("COUPON",2)/COLL_PREV_BAL(2)*(COLL_PREV_BAL(2)-BBAL
("A3")))/(COLL_PREV_BAL-BBAL("A1","A2","A3")) );
    DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
    Delay 0 Dated 20031204 Next 20040125
    (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 1.035 ELSE 0.69 ))
    0 999
!

```

```

Tranche "M2" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 43125000.00 at 2.87  FREQ M FLOAT RESET M _
    COUPONCAP 30360 NONE ( 1200*(COLL_I_MISC
("COUPON",1)/COLL_PREV_BAL(1)*(COLL_PREV_BAL(1)-BBAL("A1","A2"))
+COLL_I_MISC("COUPON",2)/COLL_PREV_BAL(2)*(COLL_PREV_BAL(2)-BBAL
("A3")))/(COLL_PREV_BAL-BBAL("A1","A2","A3")) );
    DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
    Delay 0 Dated 20031204 Next 20040125
    (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 2.625 ELSE 1.75 ))
    0 999
!

```

```

Tranche "M3" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 11250000.00 at 3.2  FREQ M FLOAT RESET M _
    COUPONCAP 30360 NONE ( 1200*(COLL_I_MISC
("COUPON",1)/COLL_PREV_BAL(1)*(COLL_PREV_BAL(1)-BBAL("A1","A2"))
+COLL_I_MISC("COUPON",2)/COLL_PREV_BAL(2)*(COLL_PREV_BAL(2)-BBAL
("A3")))/(COLL_PREV_BAL-BBAL("A1","A2","A3")) );
    DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
    Delay 0 Dated 20031204 Next 20040125
    (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 3.12 ELSE 2.08 ))
    0 999
!

```

```

Tranche "M4" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 11250000.00 at 4.47  FREQ M FLOAT RESET M _
    COUPONCAP 30360 NONE ( 1200*(COLL_I_MISC
("COUPON",1)/COLL_PREV_BAL(1)*(COLL_PREV_BAL(1)-BBAL("A1","A2"))
+COLL_I_MISC("COUPON",2)/COLL_PREV_BAL(2)*(COLL_PREV_BAL(2)-BBAL
("A3")))/(COLL_PREV_BAL-BBAL("A1","A2","A3")) );
    DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
    Delay 0 Dated 20031204 Next 20040125
    (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 5.025 ELSE 3.35 ))
    0 999
!

```

```

Tranche "M5" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 11250000.00 at 4.87  FREQ M FLOAT RESET M _
    COUPONCAP 30360 NONE ( 1200*(COLL_I_MISC
("COUPON",1)/COLL_PREV_BAL(1)*(COLL_PREV_BAL(1)-BBAL("A1","A2"))
+COLL_I_MISC("COUPON",2)/COLL_PREV_BAL(2)*(COLL_PREV_BAL(2)-BBAL
("A3")))/(COLL_PREV_BAL-BBAL("A1","A2","A3")) );
    DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
    Delay 0 Dated 20031204 Next 20040125
    (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 5.625 ELSE 3.75 ))
    0 999
!

```

```

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

```



```

      Block 9375000.00 at 4.87 FREQ M FLOAT RESET M
      COUPONCAP 30360 NONE ( 1200*(COLL_I MISC
("COUPON",1)/COLL_PREV_BAL(1)*(COLL_PREV_BAL(1)-BBAL("A1","A2"))
+COLL_I_MISC("COUPON",2)/COLL_PREV_BAL(2)*(COLL_PREV_BAL(2)-BBAL
("A3")))/(COLL_PREV_BAL-BBAL("A1","A2","A3")) );
      DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
      Delay 0 Dated 20031204 Next 20040125
      (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1") / #OrigCollBal) < 10%)
THEN 5.625 ELSE 3.75 ))
      0 999

```

```

!
Tranche "R" JUN_RES
      Block 750000000.00 at 0 NOTIONAL WITH GROUP 0 SURPLUS
      DAYCOUNT 30360 BUSINESS_DAY NONE
      FREQ M Delay 0 Dated 20031225 Next 20040125

```

```

!
Tranche "OC" JUN_OC_RES
      Block 19125000.00 at 0
      DAYCOUNT 30360 BUSINESS_DAY NONE
      FREQ M Delay 0 Dated 20031225 Next 20040125

```

```

!
Tranche "P" JUN_PEN_NO
      Block 750000000.00 at 0 NOTIONAL WITH GROUP 0
      DAYCOUNT 30360 BUSINESS_DAY NONE
      FREQ M Delay 0 Dated 20031225 Next 20040125

```

```

!
      Tranche "#OC" SYMVAR
      Tranche "#SpecOCTarg" SYMVAR

```

```

!
DEFINE PSEUDO_TRANCHE COLLAT
      Delay 24 Dated 20031201 Next 20040125 Settle 20031204
DEFINE PSEUDO_TRANCHE COLLAT GROUP 1
      Delay 24 Dated 20031201 Next 20040125 Settle 20031204
DEFINE PSEUDO_TRANCHE COLLAT GROUP 2
      Delay 24 Dated 20031201 Next 20040125 Settle 20031204

```

```

!
      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"
!
      CLASS "SNR_11" NO_BUILD_TRANCHE
      SHORTFALL_PAYBACK COUPONCAP TRUE
      SHORTFALL_EARN_INT COUPONCAP TRUE

```

```

      = "A1"
CLASS "SNR_12" NO_BUILD_TRANCHE _
               SHORTFALL_PAYBACK COUPONCAP TRUE _
               SHORTFALL_EARN_INT COUPONCAP TRUE _
               = "A2"
CLASS "SNR_2" NO_BUILD_TRANCHE _
              SHORTFALL_PAYBACK COUPONCAP TRUE _
              SHORTFALL_EARN_INT COUPONCAP TRUE _
              = "A3"
CLASS "MEZ1" NO_BUILD_TRANCHE _
             SHORTFALL_PAYBACK COUPONCAP TRUE _
             SHORTFALL_EARN_INT COUPONCAP TRUE _
             = "M1"
CLASS "MEZ2" NO_BUILD_TRANCHE _
             SHORTFALL_PAYBACK COUPONCAP TRUE _
             SHORTFALL_EARN_INT COUPONCAP TRUE _
             = "M2"
CLASS "MEZ3" NO_BUILD_TRANCHE _
             SHORTFALL_PAYBACK COUPONCAP TRUE _
             SHORTFALL_EARN_INT COUPONCAP TRUE _
             = "M3"
CLASS "MEZ4" NO_BUILD_TRANCHE _
             SHORTFALL_PAYBACK COUPONCAP TRUE _
             SHORTFALL_EARN_INT COUPONCAP TRUE _
             = "M4"
CLASS "MEZ5" NO_BUILD_TRANCHE _
             SHORTFALL_PAYBACK COUPONCAP TRUE _
             SHORTFALL_EARN_INT COUPONCAP TRUE _
             = "M5"
CLASS "MEZ6" NO_BUILD_TRANCHE _
             SHORTFALL_PAYBACK COUPONCAP TRUE _
             SHORTFALL_EARN_INT COUPONCAP TRUE _
             = "M6"
CLASS "RESID" = "R#1" "OC#1" "P#1"
CLASS "SNR_1" DISTRIB_CLASS PRORATA ALLOCATION _
              = "SNR_11" "SNR_12"
CLASS "SNR" DISTRIB_CLASS PRORATA WRITEDOWN_BAL PRORATA ALLOCATION _
            = "SNR_1" "SNR_2"
!
!
CLASS "ROOT" _
          WRITEDOWN_BAL RULES _
          DISTRIB_CLASS RULES _
          SHORTFALL_PAYBACK PRINCIPAL_LOSS TRUE _
          SHORTFALL_EARN_INT INTEREST TRUE _
          = "SNR" "MEZ1" "MEZ2" "MEZ3" "MEZ4" "MEZ5" "MEZ6"
"RESID"
!
  DEFINE PSEUDO_TRANCHE CLASS "SNR" Delay 24 Dated 20031204
Next 20040125 DAYCOUNT 30360 BUSINESS_DAY NONE
!
  DEFINE PSEUDO_TRANCHE CLASS "SNR_1" Delay 0 Dated 20031204
Next 20040125 DAYCOUNT 30360 BUSINESS_DAY NONE
!
!
  CROSSOVER When 0
!
  TRIGGER "StepUp-CumLoss" _
    FULL_NAME "Step Up Cumulative Loss Trigger" _
    ORIG_TESTVAL 0.000% _
    TESTVAL ( #TrigCumLossFrac); _

```

```

    ORIG_TARGETVAL 3.25% _
    TARGETVAL      (#CumLossShft); _
    TRIGVAL        LODIFF
!
TRIGGER "StepUp-DlqEnh" _
    FULL_NAME      "Step Up Enhancement Delinquency Trigger" _
    ORIG_TESTVAL   0.000% _
    TESTVAL        ( #TrigEnhFrac); _
    ORIG_TARGETVAL 20.8% _
    TARGETVAL      (#ReqPerc); _
    TRIGVAL        LODIFF
!
TRIGGER "STEPUP_TRIGGER" _
    FULL_NAME      "Step Up Trigger" _
    DEFINITION "A Step Up Trigger exists, if _
; (1) a percentage calculated as the quotient of the amount of
cumulative
realized losses divided by the original collateral balance exceeds the
target defined by a schedule; _
        Month <=          %; _
        48                3.25%; _
        60                5.25%; _
        72                6.75%; _
        84                7.5%; _
        360               7.75%; _
or; (2) the aggregate principal balance of all delinquent loans * 1_
as a percentage of the respective collateral balance exceeds : _
0.395 * the Senior Enhancement Percentage."
    IMPACT         "If a Step Up Trigger is in effect the OC target will
change to _
the last value before the trigger occurred if a stepdown has _
occurred. It has no effect if a stepdown has not occurred."
    TRIGVAL FORMULA ( min(TRIGGER("StepUp-CumLoss", "TRIGVAL"),
TRIGGER("StepUp-DlqEnh", "TRIGVAL")));
!
    OPTIONAL REDEMPTION: "CLEANUP" _
                        COLL_FRAC 10% _
                        PRICE_P ( COLL_BAL );
!
!
INTEREST_SHORTFALL FULL_PREPAY    Compensate Pro_rata _
                  PARTIAL_PREPAY Compensate Pro_rata _
                  LOSS            Compensate Pro_rata _
!
TRANCHE MISCINFO
A1          RATING FT "AAA" MD "Aaa" SP "AAA"
A2          RATING FT "AAA" MD "Aaa" SP "AAA"
A3          RATING FT "AAA" MD "Aaa" SP "AAA"
M1          RATING FT "AA" MD "Aa2" SP "AA"
M2          RATING FT "A" MD "A2" SP "A"
M3          RATING FT "A-" MD "A3" SP "A-"
M4          RATING FT "BBB+" MD "Baa1" SP "BBB+"
M5          RATING FT "BBB" MD "Baa2" SP "BBB"
M6          RATING MD "Baa3" SP "BBB-"
R           RATING MD "NA"
OC          RATING MD "NA"
P           RATING MD "NA"
!
DEFINE MACRO BLOCK #SNR_Int =
{

```

```

-----
      from : CLASS ( "SNR" )
      pay : CLASS INTEREST PRO_RATA ( "SNR_1"; "SNR_2" )
-----

      from : CLASS ( "SNR_1" )
      pay : CLASS INTEREST PRO_RATA ( "SNR_11"; "SNR_12" )
-----
}
DEFINE MACRO BLOCK #SNR_InS =
{
-----
      from : CLASS ( "SNR" )
      pay : CLASS INTSHORT PRO_RATA ( "SNR_1"; "SNR_2" )
-----

      from : CLASS ( "SNR_1" )
      pay : CLASS INTSHORT PRO_RATA ( "SNR_11"; "SNR_12" )
-----
}
DEFINE MACRO BLOCK #SNR_Prn[1] =
{
-----
      when : IS_TRUE( {#1} )
      calculate : #SeniorPrinc = #PrincPmt / #DistribAmt *
#ClassSNRPDA
      calculate : #SeniorXtraP = #ClassSNRPDA - #SeniorPrinc
!
      calculate : #SeniorPDA1 = MAX( 0, MIN( #ClassSNR_1PDA,
#SenDistribAmt1 ) ) + _
MIN( #ClassSNR_1PDADefic,
#ClassSNR_1PDADefic/#TotalSenPDADefic * #TotalExcessDistrib)
      calculate : #SeniorPDA2 = MAX( 0, MIN( #ClassSNR_2PDA,
#SenDistribAmt2 ) ) + _
MIN( #ClassSNR_2PDADefic,
#ClassSNR_2PDADefic/#TotalSenPDADefic * #TotalExcessDistrib)
-----
      from : SUBACCOUNT ( #SeniorPDA1, CLASS "SNR" )
      pay : CLASS BALANCE SEQUENTIAL ( "SNR_1" )
-----

      from : SUBACCOUNT ( #SeniorPDA2, CLASS "SNR" )
      pay : CLASS BALANCE SEQUENTIAL ( "SNR_2" )
-----

      from : CLASS ( "SNR" )
      pay : CLASS BALANCE PRO_RATA ( "SNR_1"; "SNR_2" )
-----
!
      calculate : #P_SNR_1 = BBAL("SNR_11","SNR_12") - BBAL("SNR_1")
-----

      calculate : #P_A1 = BBAL("A1") / BBAL("A1","A2") * #P_SNR_1
      calculate : #P_A2 = BBAL("A2") / BBAL("A1","A2") * #P_SNR_1
-----

      from : CLASS ( "SNR_1" )
      pay : CLASS INTEREST SEQUENTIAL ( "SNR_11" )
      pay : CLASS INTSHORT SEQUENTIAL ( "SNR_11" )
-----

      from : CLASS ( "SNR_1" )
      from : SUBACCOUNT ( #P_SNR_1 )
subject to : CEILING ( #P_A1 )
      pay : CLASS BALANCE SEQUENTIAL ( "SNR_11" )
-----

```

```

      from : CLASS ( "SNR_1" )
      pay : CLASS INTEREST SEQUENTIAL ( "SNR_12" )
      pay : CLASS INTSHORT SEQUENTIAL ( "SNR_12" )
-----
      from : CLASS ( "SNR_1" )
      from : SUBACCOUNT ( #P_SNR_1 )
subject to : CEILING ( #P_A2 )
      pay : CLASS BALANCE SEQUENTIAL ( "SNR_12" )
-----
!
-----
      from : CLASS ( "SNR_11" )
      pay : SEQUENTIAL ( "A1#1" )
-----
      from : CLASS ( "SNR_12" )
      pay : SEQUENTIAL ( "A2#1" )
-----
      from : CLASS ( "SNR_2" )
      pay : SEQUENTIAL ( "A3#1" )
-----
}
DEFINE MACRO BLOCK #MEZ1_Prn =
{
-----
      from : CLASS ( "MEZ1" )
      pay : SEQUENTIAL ( "M1#1" )
-----
}
DEFINE MACRO BLOCK #MEZ2_Prn =
{
-----
      from : CLASS ( "MEZ2" )
      pay : SEQUENTIAL ( "M2#1" )
-----
}
DEFINE MACRO BLOCK #MEZ3_Prn =
{
-----
      from : CLASS ( "MEZ3" )
      pay : SEQUENTIAL ( "M3#1" )
-----
}
DEFINE MACRO BLOCK #MEZ4_Prn =
{
-----
      from : CLASS ( "MEZ4" )
      pay : SEQUENTIAL ( "M4#1" )
-----
}
DEFINE MACRO BLOCK #MEZ5_Prn =
{
-----
      from : CLASS ( "MEZ5" )
      pay : SEQUENTIAL ( "M5#1" )
-----
}
DEFINE MACRO BLOCK #MEZ6_Prn =
{
-----
      from : CLASS ( "MEZ6" )
      pay : SEQUENTIAL ( "M6#1" )

```

```

}
!
-----
CMO Block Payment Rules
-----
calculate : #HedgePaySave = 0.00
-----
calculate : #PrincFrac1      = COLL_P(1) / COLL_P
calculate : #PrincFrac2      = COLL_P(2) / COLL_P
!
calculate : #XtraPFrac1      = COLL_P(1) / COLL_P
calculate : #XtraPFrac2      = COLL_P(2) / COLL_P
!
calculate : #Princ           = COLL_P
!
calculate : #Interest        = COLL_I
!
calculate : #PrevSpecOC      = #SpecOCTarg
!
calculate : #CurrentOC       = MAX( 0, COLL_BAL - (BBAL("A1#1",
"A2#1", "A3#1", "M1#1", "M2#1", "M3#1", "M4#1", "M5#1", "M6#1") -
#Princ))
!
calculate : #XSSpread        = MAX( 0, #Interest -
OPTIMAL_INTPMT("ROOT") - INTSHORT_ACCUM("SNR") + 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 -
OPTIMAL_INTPMT("ROOT") - INTSHORT_ACCUM("SNR") + #OCSurplus +
COUPONCAP_SHORTFALL("ROOT"))
!
calculate : #SubDefic        = MAX ( 0, ( BBAL("ROOT") - BBAL(
"OC#1" ) - #Princ ) - COLL_BAL )
!
calculate : #AddPrinc        = MIN( #XSIntRem, #SubDefic )
calculate : #XSIntRem        = MAX( 0, #XSIntRem - #AddPrinc )
!
calculate : #XtraPDA         = MIN( #OCDeficiency, #XSIntRem )
calculate : #XSIntRem        = MAX( 0, #XSIntRem - #XtraPDA )

```

```

!
!   calculate : #DistribAmt          = #PrincPmt + #AddPrinc + #XtraPDA
!
!   calculate : #SenDistribAmt1      = #PrincPmt * #PrincFrac1 +
(#DistribAmt - #PrincPmt) * #XtraPFrac1
!   calculate : #SenDistribAmt2      = #PrincPmt * #PrincFrac2 +
(#DistribAmt - #PrincPmt) * #XtraPFrac2
!
!   calculate : #FloorOCTotal1       = #OrigCollBal1/#OrigCollBal *
#FloorOCTotal
!   calculate : #FloorOCTotal2       = #OrigCollBal2/#OrigCollBal *
#FloorOCTotal
!
!   calculate : #ClassSNR_1PDA       = BBAL("A1", "A2")
- MIN(COLL_BAL(1) -
#FloorOCTotal1, #SNRTargPct * COLL_BAL(1))
!   calculate : #ClassSNR_1PDA       = MAX( 0.0, MIN(BBAL("A1", "A2"),
#ClassSNR_1PDA ))
!   calculate : #ClassSNR_2PDA       = BBAL("A3")
- MIN(COLL_BAL(2) -
#FloorOCTotal2, #SNRTargPct * COLL_BAL(2))
!   calculate : #ClassSNR_2PDA       = MAX( 0.0, MIN(BBAL("A3"),
#ClassSNR_2PDA ))
!
!   calculate : #ClassSNR_1PDADefic  = MAX( 0.0, #ClassSNR_1PDA -
#SenDistribAmt1 )
!   calculate : #ClassSNR_2PDADefic  = MAX( 0.0, #ClassSNR_2PDA -
#SenDistribAmt2 )
!   calculate : #TotalSenPDADefic    = #ClassSNR_1PDADefic + #ClassSNR_
2PDADefic
!
!   calculate : #ExcessDistrib1      = MAX( 0.0, #SenDistribAmt1 -
#ClassSNR_1PDA )
!   calculate : #ExcessDistrib2      = MAX( 0.0, #SenDistribAmt2 -
#ClassSNR_2PDA )
!   calculate : #TotalExcessDistrib  = #ExcessDistrib1 + #ExcessDistrib2
!
!   calculate : #ExcessBalance1      = BBAL("SNR_1") - MIN(
#SenDistribAmt1, #ClassSNR_1PDA )
!   calculate : #ExcessBalance2      = BBAL("SNR_2") - MIN(
#SenDistribAmt2, #ClassSNR_2PDA )
!   calculate : #ExcessBalance       = #ExcessBalance1 + #ExcessBalance2
!
!   calculate : #ExcessDistrib       = 0
!
!   calculate : #ClassSNRPDA         = #ClassSNR_1PDA + #ClassSNR_2PDA +
#ExcessDistrib
!   calculate : #ClassSNRPDA         = MIN( #ClassSNRPDA, #DistribAmt )
!
!!!***** END OF SENIOR ENHANCEMENT PCT CALCULATION *****
!
!   calculate : #SenEnhancePct       = (COLL_BAL - (BBAL("SNR") -
#ClassSNRPDA )) / COLL_BAL
!
!   calculate : #StepDownBal         = (#SenEnhancePct - #SpecSenEnhPct)
+ 1E-8 GE 0.00
!
!   calculate : #StepDown            = #StepDown OR ( BBAL("SNR") LT
0.01 ) OR ( #StepDownDatePass AND #StepDownBal )
!
!   calculate : #ReqPerc             = 0.44 * (COLL_PREV_BAL - BBAL

```

```

("SNR") ) / COLL_PREV_BAL
!
! calculate : #TrigEnhFrac = 1 * AVG_COLL("RATE",-1,2,1)
!
! calculate : #CumLossShift = LOOKUP_TBL( "STEP", CURMONTH
, "OC_CUMLOSS0", "MONTH", "OC_CUMLOSS_FRAC0" )
! calculate : #TrigCumLossFrac = DELINQ_LOSS_ACCUM / #OrigCollBal
!
! calculate : #TrigEvent = TRIGGER("STEPUP_TRIGGER")
!
! calculate : #TrigOCTargPost = #PrevSpecOC
!
! calculate : #SpecOCTarg = IF #StepDown _
! THEN IF #TrigEvent _
! THEN MAX( MIN( #InitOCTarg,
#StepOCTarg ) , #TrigOCTargPost, #FloorOCTotal ) _
! ELSE MAX( MIN( #InitOCTarg,
#StepOCTarg ) , #FloorOCTotal ) -
#FloorOCTotal )
!
! calculate : #SpecOCTarg = MIN( #SpecOCTarg, COLL_BAL )
!
! calculate : #SpecOCTarg = #Octval
!
! calculate : #OCDeficiency = MAX(0, #SpecOCTarg - #CurrentOC)
!
! calculate : #OCSurplus = MINMAX(0, #CurrentOC -
#SpecOCTarg, COLL_P)
!
! calculate : #PrincPmt = MAX(0, COLL_P - #OCSurplus)
!
!
! calculate : #XSIntRem = MAX( 0, #Interest -
OPTIMAL_INTPMT("ROOT") - INTSHORT_ACCUM("SNR") + #OCSurplus +
COUPONCAP_SHORTFALL("ROOT"))
!
! calculate : #SubDefic = MAX ( 0, ( BBAL("ROOT") - BBAL(
"OC#1" ) - #Princ ) - COLL_BAL )
!
! calculate : #AddPrinc = MIN( #XSIntRem, #SubDefic )
! calculate : #XSIntRem = MAX( 0, #XSIntRem - #AddPrinc )
!
! calculate : #XtraPDA = MIN( #OCDeficiency, #XSIntRem )
! calculate : #XSIntRem = MAX( 0, #XSIntRem - #XtraPDA )
!
! calculate : #DistribAmt = #PrincPmt + #AddPrinc + #XtraPDA
!
! calculate : #SenDistribAmt1 = #PrincPmt * #PrincFrac1 +
(#DistribAmt - #PrincPmt) * #XtraPFrac1
! calculate : #SenDistribAmt2 = #PrincPmt * #PrincFrac2 +
(#DistribAmt - #PrincPmt) * #XtraPFrac2
!
! calculate : #FloorOCTotal1 = #OrigCollBal1/#OrigCollBal *
#FloorOCTotal
! calculate : #FloorOCTotal2 = #OrigCollBal2/#OrigCollBal *
#FloorOCTotal
!
! calculate : #ClassSNR_1PDA = IF (#TrigEvent OR (#StepDown EQ
0.0)) _
! THEN #SenDistribAmt1

```



```

ELSE BBAL("A1", "A2") _
- MIN(COLL_BAL(1) -
#FloorOCTotal1, #SNRTargPct * COLL_BAL(1))
calculate : #ClassSNR_1PDA = MAX( 0.0, MIN(BBAL("A1", "A2"),
#ClassSNR_1PDA ))
calculate : #ClassSNR_2PDA = IF (#TrigEvent OR (#StepDown EQ
0.0)) _
THEN #SenDistribAmt2 _
ELSE BBAL("A3")
- MIN(COLL_BAL(2) -
#FloorOCTotal2, #SNRTargPct * COLL_BAL(2))
calculate : #ClassSNR_2PDA = MAX( 0.0, MIN(BBAL("A3"),
#ClassSNR_2PDA ))
!
calculate : #ClassSNR_1PDADefic = MAX( 0.0, #ClassSNR_1PDA -
#SenDistribAmt1 )
calculate : #ClassSNR_2PDADefic = MAX( 0.0, #ClassSNR_2PDA -
#SenDistribAmt2 )
calculate : #TotalSenPDADefic = #ClassSNR_1PDADefic + #ClassSNR_
2PDADefic
!
calculate : #ExcessDistrib1 = MAX( 0.0, #SenDistribAmt1 -
#ClassSNR_1PDA )
calculate : #ExcessDistrib2 = MAX( 0.0, #SenDistribAmt2 -
#ClassSNR_2PDA )
calculate : #TotalExcessDistrib = #ExcessDistrib1 + #ExcessDistrib2
!
calculate : #ExcessBalance1 = BBAL("SNR_1") - MIN(
#SenDistribAmt1, #ClassSNR_1PDA )
calculate : #ExcessBalance2 = BBAL("SNR_2") - MIN(
#SenDistribAmt2, #ClassSNR_2PDA )
calculate : #ExcessBalance = #ExcessBalance1 + #ExcessBalance2
!
calculate : #ExcessDistrib = IF (#TrigEvent OR (#StepDown EQ
0.0)) _
THEN MIN( #ExcessBalance,
#ExcessDistrib1 + #ExcessDistrib2 ) _
ELSE 0
!
calculate : #ClassSNRPDA = #ClassSNR_1PDA + #ClassSNR_2PDA +
#ExcessDistrib
calculate : #ClassSNRPDA = MIN( #ClassSNRPDA, #DistribAmt )
!
calculate : #ClassMEZ1PDA = IF (#TrigEvent OR (#StepDown EQ
0.0)) _
THEN #DistribAmt - #ClassSNRPDA
-
ELSE BBAL("A1", "A2", "A3",
- MIN(COLL_BAL - #FloorOCTotal,
#MEZ1TargPct * COLL_BAL)
calculate : #ClassMEZ1PDA = MAX( 0.0, MIN(BBAL("M1"),
#ClassMEZ1PDA ))
calculate : #ClassMEZ1PDA = MAX( 0, MIN( #ClassMEZ1PDA,
#DistribAmt - #ClassSNRPDA ) )
!
!
calculate : #ClassMEZ2PDA = IF (#TrigEvent OR (#StepDown EQ
0.0)) _
THEN #DistribAmt - #ClassSNRPDA
- #ClassMEZ1PDA _

```

```

ELSE BBAL("A1", "A2", "A3",
"M1", "M2") - #ClassSNRPDA - #ClassMEZ1PDA
- MIN(COLL_BAL - #FloorOCTotal,
#MEZ2TargPct * COLL_BAL)
calculate : #ClassMEZ2PDA = MAX( 0.0, MIN(BBAL("M2"),
#ClassMEZ2PDA ))
calculate : #ClassMEZ2PDA = MAX( 0, MIN( #ClassMEZ2PDA,
#DistribAmt - #ClassSNRPDA - #ClassMEZ1PDA ) )
!
!
calculate : #ClassMEZ3PDA = IF (#TrigEvent OR (#StepDown EQ
0.0)) _
THEN #DistribAmt - #ClassSNRPDA
- #ClassMEZ1PDA - #ClassMEZ2PDA _
ELSE BBAL("A1", "A2", "A3",
"M1", "M2", "M3") - #ClassSNRPDA - #ClassMEZ1PDA - #ClassMEZ2PDA _
- MIN(COLL_BAL - #FloorOCTotal,
#MEZ3TargPct * COLL_BAL)
calculate : #ClassMEZ3PDA = MAX( 0.0, MIN(BBAL("M3"),
#ClassMEZ3PDA ))
calculate : #ClassMEZ3PDA = MAX( 0, MIN( #ClassMEZ3PDA,
#DistribAmt - #ClassSNRPDA - #ClassMEZ1PDA - #ClassMEZ2PDA ) )
!
!
calculate : #ClassMEZ4PDA = IF (#TrigEvent OR (#StepDown EQ
0.0)) _
THEN #DistribAmt - #ClassSNRPDA
- #ClassMEZ1PDA - #ClassMEZ2PDA - #ClassMEZ3PDA _
ELSE BBAL("A1", "A2", "A3",
"M1", "M2", "M3", "M4") - #ClassSNRPDA - #ClassMEZ1PDA - #ClassMEZ2PDA -
#ClassMEZ3PDA _
- MIN(COLL_BAL - #FloorOCTotal,
#MEZ4TargPct * COLL_BAL)
calculate : #ClassMEZ4PDA = MAX( 0.0, MIN(BBAL("M4"),
#ClassMEZ4PDA ))
calculate : #ClassMEZ4PDA = MAX( 0, MIN( #ClassMEZ4PDA,
#DistribAmt - #ClassSNRPDA - #ClassMEZ1PDA - #ClassMEZ2PDA -
#ClassMEZ3PDA ) )
!
!
calculate : #ClassMEZ5PDA = IF (#TrigEvent OR (#StepDown EQ
0.0)) _
THEN #DistribAmt - #ClassSNRPDA
- #ClassMEZ1PDA - #ClassMEZ2PDA - #ClassMEZ3PDA - #ClassMEZ4PDA _
ELSE BBAL("A1", "A2", "A3",
"M1", "M2", "M3", "M4", "M5") - #ClassSNRPDA - #ClassMEZ1PDA -
#ClassMEZ2PDA - #ClassMEZ3PDA - #ClassMEZ4PDA _
- MIN(COLL_BAL - #FloorOCTotal,
#MEZ5TargPct * COLL_BAL)
calculate : #ClassMEZ5PDA = MAX( 0.0, MIN(BBAL("M5"),
#ClassMEZ5PDA ))
calculate : #ClassMEZ5PDA = MAX( 0, MIN( #ClassMEZ5PDA,
#DistribAmt - #ClassSNRPDA - #ClassMEZ1PDA - #ClassMEZ2PDA -
#ClassMEZ3PDA - #ClassMEZ4PDA ) )
!
!
calculate : #ClassMEZ6PDA = IF (#TrigEvent OR (#StepDown EQ
0.0)) _
THEN #DistribAmt - #ClassSNRPDA
- #ClassMEZ1PDA - #ClassMEZ2PDA - #ClassMEZ3PDA - #ClassMEZ4PDA -
#ClassMEZ5PDA _

```

```

ELSE BBAL("A1", "A2", "A3",
"M1", "M2", "M3", "M4", "M5", "M6") - #ClassSNRPDA - #ClassMEZ1PDA -
#ClassMEZ2PDA - #ClassMEZ3PDA - #ClassMEZ4PDA - #ClassMEZ5PDA -
MIN(COLL_BAL - #FloorOCTotal,
#MEZ6TargPct * COLL_BAL)
calculate : #ClassMEZ6PDA = MAX( 0.0, MIN(BBAL("M6"),
#ClassMEZ6PDA ))
calculate : #ClassMEZ6PDA = MAX( 0, MIN( #ClassMEZ6PDA,
#DistribAmt - #ClassSNRPDA - #ClassMEZ1PDA - #ClassMEZ2PDA -
#ClassMEZ3PDA - #ClassMEZ4PDA - #ClassMEZ5PDA ) )
!
!
calculate : "SNR"
NO_CHECK CUSTOM AMOUNT = #ClassSNRPDA
!
calculate : "MEZ1"
NO_CHECK CUSTOM AMOUNT = #ClassMEZ1PDA
!
calculate : "MEZ2"
NO_CHECK CUSTOM AMOUNT = #ClassMEZ2PDA
!
calculate : "MEZ3"
NO_CHECK CUSTOM AMOUNT = #ClassMEZ3PDA
!
calculate : "MEZ4"
NO_CHECK CUSTOM AMOUNT = #ClassMEZ4PDA
!
calculate : "MEZ5"
NO_CHECK CUSTOM AMOUNT = #ClassMEZ5PDA
!
calculate : "MEZ6"
NO_CHECK CUSTOM AMOUNT = #ClassMEZ6PDA
!
calculate : "RESID"
NO_CHECK CUSTOM AMOUNT = MAX(0, #Princ - OPTIMAL_PRINCPMT
("SNR", "MEZ1", "MEZ2", "MEZ3", "MEZ4", "MEZ5", "MEZ6"))
!
calculate : #M1BegBal = BBAL("M1")
calculate : #M2BegBal = BBAL("M2")
calculate : #M3BegBal = BBAL("M3")
calculate : #M4BegBal = BBAL("M4")
calculate : #M5BegBal = BBAL("M5")
calculate : #M6BegBal = BBAL("M6")

```

```

-----
pay : CLASS INTEREST PRO_RATA ( "SNR" )
-----
{#SNR_Int}
-----
pay : CLASS INTSHORT PRO_RATA ( "SNR" )
-----
{#SNR_InS}
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "SNR" )
-----
{#SNR_Prn}{1}
-----
-----

```

```

        pay : CLASS INTEREST PRO_RATE ( "MEZ1" )
        pay : CLASS INTSHORT PRO_RATE ( "MEZ1" )
        pay : CLASS PRINCIPAL SEQUENTIAL ( "MEZ1" )
-----
{#MEZ1_Prn}
-----
        pay : CLASS PRINCESHORT_LOSS SEQUENTIAL ( "MEZ1" )
-----
        pay : CLASS INTEREST PRO_RATE ( "MEZ2" )
        pay : CLASS INTSHORT PRO_RATE ( "MEZ2" )
        pay : CLASS PRINCIPAL SEQUENTIAL ( "MEZ2" )
-----
{#MEZ2_Prn}
-----
        pay : CLASS PRINCESHORT_LOSS SEQUENTIAL ( "MEZ2" )
-----
        pay : CLASS INTEREST PRO_RATE ( "MEZ3" )
        pay : CLASS INTSHORT PRO_RATE ( "MEZ3" )
        pay : CLASS PRINCIPAL SEQUENTIAL ( "MEZ3" )
-----
{#MEZ3_Prn}
-----
        pay : CLASS PRINCESHORT_LOSS SEQUENTIAL ( "MEZ3" )
-----
        pay : CLASS INTEREST PRO_RATE ( "MEZ4" )
        pay : CLASS INTSHORT PRO_RATE ( "MEZ4" )
        pay : CLASS PRINCIPAL SEQUENTIAL ( "MEZ4" )
-----
{#MEZ4_Prn}
-----
        pay : CLASS PRINCESHORT_LOSS SEQUENTIAL ( "MEZ4" )
-----
        pay : CLASS INTEREST PRO_RATE ( "MEZ5" )
        pay : CLASS INTSHORT PRO_RATE ( "MEZ5" )
        pay : CLASS PRINCIPAL SEQUENTIAL ( "MEZ5" )
-----
{#MEZ5_Prn}
-----
        pay : CLASS PRINCESHORT_LOSS SEQUENTIAL ( "MEZ5" )
-----
        pay : CLASS INTEREST PRO_RATE ( "MEZ6" )
        pay : CLASS INTSHORT PRO_RATE ( "MEZ6" )
        pay : CLASS PRINCIPAL SEQUENTIAL ( "MEZ6" )
-----
{#MEZ6_Prn}
-----
        pay : CLASS PRINCESHORT_LOSS SEQUENTIAL ( "MEZ6" )
-----
        from : CLASS ( "ROOT" )
        pay : CLASS COUPONCAP_SHORT PRO_RATE ( "SNR_12" ; "SNR_2" )
-----
        from : CLASS ( "ROOT" )
        pay : CLASS COUPONCAP_SHORT PRO_RATE ( "MEZ1" )
-----

```

```

      from : CLASS ( "ROOT" )
      pay : CLASS COUPONCAP_SHORT PRO_RATA ( "MEZ2" )
-----
      from : CLASS ( "ROOT" )
      pay : CLASS COUPONCAP_SHORT PRO_RATA ( "MEZ3" )
-----
      from : CLASS ( "ROOT" )
      pay : CLASS COUPONCAP_SHORT PRO_RATA ( "MEZ4" )
-----
      from : CLASS ( "ROOT" )
      pay : CLASS COUPONCAP_SHORT PRO_RATA ( "MEZ5" )
-----
      from : CLASS ( "ROOT" )
      pay : CLASS COUPONCAP_SHORT PRO_RATA ( "MEZ6" )
-----
      from : HEDGE ("CAP")
      subject to : CEILING ((HEDGE ("CAP", "OPTIMAL_PMT") - HEDGE ("CAP",
"ACTUAL_PMT"))))
      pay : CLASS COUPONCAP_SHORT PRO_RATA("SNR_12" ; "SNR_2" ;
"MEZ1" ; "MEZ2" ; "MEZ3" ; "MEZ4" ; "MEZ5" ; "MEZ6")
-----
      from : HEDGE ("CAP2")
      subject to : CEILING ((HEDGE ("CAP2", "OPTIMAL_PMT") - HEDGE ("CAP2",
"ACTUAL_PMT"))))
      pay : CLASS COUPONCAP_SHORT PRO_RATA("SNR_12")
-----
      from : HEDGE ("CAP3")
      subject to : CEILING ((HEDGE ("CAP3", "OPTIMAL_PMT") - HEDGE ("CAP3",
"ACTUAL_PMT"))))
      pay : CLASS COUPONCAP_SHORT PRO_RATA("SNR_2")
-----
      calculate : #MEZ1_BAL_CCS = #M1BegBal *
(COUPONCAP_ACCUM_SHORTFALL("MEZ1") GT .01)
      calculate : #MEZ2_BAL_CCS = #M2BegBal *
(COUPONCAP_ACCUM_SHORTFALL("MEZ2") GT .01)
      calculate : #MEZ3_BAL_CCS = #M3BegBal *
(COUPONCAP_ACCUM_SHORTFALL("MEZ3") GT .01)
      calculate : #MEZ4_BAL_CCS = #M4BegBal *
(COUPONCAP_ACCUM_SHORTFALL("MEZ4") GT .01)
      calculate : #MEZ5_BAL_CCS = #M5BegBal *
(COUPONCAP_ACCUM_SHORTFALL("MEZ5") GT .01)
      calculate : #MEZ6_BAL_CCS = #M6BegBal *
(COUPONCAP_ACCUM_SHORTFALL("MEZ6") GT .01)
-----
      from : HEDGE ("CAP4")
      subject to : CEILING ((HEDGE ("CAP4", "OPTIMAL_PMT") - HEDGE ("CAP4",
"ACTUAL_PMT"))))
      subject to : PROPORTION ( (#MEZ1_BAL_CCS); (#MEZ2_BAL_CCS); (#MEZ3
_BAL_CCS); (#MEZ4_BAL_CCS); (#MEZ5_BAL_CCS); (#MEZ6_BAL_CCS) )
      pay : CLASS COUPONCAP_SHORT ASIS CONCURRENT("MEZ1" ; "MEZ2" ;
"MEZ3" ; "MEZ4" ; "MEZ5" ; "MEZ6")
-----
      from : HEDGE ("CAP4")
      subject to : CEILING ((HEDGE ("CAP4", "OPTIMAL_PMT") - HEDGE ("CAP4",
"ACTUAL_PMT"))))
      pay : CLASS COUPONCAP_SHORT PRO_RATA("MEZ1" ; "MEZ2" ;
"MEZ3" ; "MEZ4" ; "MEZ5" ; "MEZ6")
-----
      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 : CLASS PRINCIPAL SEQUENTIAL ( "RESID" )
pay : AS_INTEREST ( "OC#1" )
-----
pay : SEQUENTIAL ( "OC#1" )
-----
calculate : #WriteDown = MAX(0.0, BBAL("A1#1","A2#1","A3#1","M1#
1","M2#1","M3#1","M4#1","M5#1","M6#1","OC#1") - COLL_BAL)
-----
from : SUBACCOUNT ( #Writedown )
pay : WRITEDOWN PRO_RATA ( "OC#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 : #WriteDown = IF (BBAL("A1#1","A2#1","A3#1") > COLL_BAL)
THEN MAX(0.0, BBAL("A1#1","A2#1") - COLL_BAL(1)) ELSE 0.0
-----
from : SUBACCOUNT ( #Writedown )
pay : WRITEDOWN SEQUENTIAL ( "A2#1" )
-----
calculate : #BondBal = BBAL("A1#1","A2#1","A3#1","M1#1","M2#
1","M3#1","M4#1","M5#1","M6#1")
calculate : #OC = MAX( 0, COLL_BAL - #BondBal )
calculate : #IncrOC = MAX( 0, #OC - BBAL( "OC#1" ) )
-----
pay : INCREMENT ( BALANCE "OC#1" , BY #IncrOC )
-----
!
Collateral OVER
!
! Factor --Delay--
! Type Date P/Y BV Use BV for 0

```

```

WL 20031201 9999 9999 FALSE
!
! Pool# Type Gross Current Original --Fee-- Maturity
Orig ARM Gross #mos #mos P#mos P#mos Life Reset Life
Max Look
!
Coupon Factor Balance P/Y BV P/Y BV
Term Index Margin ToRst RstPer ToRst RstPer Cap Cap
Floor Negam Back
!! BEGINNING OF COLLATERAL
M 1 "15 YR FIXED" WL 00 WAC 6.0435 (
788110.61 / 788110.61 ); 788110.61
0.5032 0.5032 177:3 177:3 180 NO_CHECK
GROUP "11F"
M 2 "15 YR FIXED" WL 00 WAC 7.0271 (
3077071.73 / 3077071.73 ); 3077071.73
0.5032 0.5032 178:2 178:2 180 NO_CHECK
GROUP "11F"
M 3 "20 YR FIXED" WL 00 WAC 6.4632 (
197299.47 / 197299.47 ); 197299.47
0.5032 0.5032 237:3 237:3 240 NO_CHECK
GROUP "11F"
M 4 "20 YR FIXED" WL 00 WAC 6.8349 (
629146.68 / 629146.68 ); 629146.68
0.5032 0.5032 239:1 239:1 240 NO_CHECK
GROUP "11F"
M 5 "20 YR FIXED" WL 00 WAC 7.1613 (
1850863.60 / 1850863.60 ); 1850863.60
0.5032 0.5032 238:2 238:2 240 NO_CHECK
GROUP "11F"
M 6 "30 YR FIXED" WL 00 WAC 7.5813 (
16737836.84 / 16737836.84 ); 16737836.84
0.5032 0.5032 358:2 358:2 360 NO_CHECK
GROUP "11F"
M 7 "30 YR FIXED" WL 00 WAC 7.1019 (
16370371.68 / 16370371.68 ); 16370371.68
0.5032 0.5032 358:2 358:2 360 NO_CHECK
GROUP "11F"
M 8 "30 YR FIXED" WL 00 WAC 7.0514 (
51755067.76 / 51755067.76 ); 51755067.76
0.5032 0.5032 358:2 358:2 360 NO_CHECK
GROUP "11F"
M 9 "2/18 ARM" WL 00 WAC 6.15 (
168279.53 / 168279.53 ); 168279.53
0.5032 0.5032 238:2 238:2 240 NO_CHECK ARM
LIBOR_6MO 6.5 23 6 SYNC_INT 12.15
1 6.15 0 0 INIT_PERCAP 2 GROUP "11A"
TEASER
M 10 "2/18 ARM" WL 00 WAC 5.95 (
120614.82 / 120614.82 ); 120614.82
0.5032 0.5032 239:1 239:1 240 NO_CHECK ARM
LIBOR_6MO 6.5 24 6 SYNC_INT 11.95
1 5.95 0 0 INIT_PERCAP 2 GROUP "11A"
TEASER
M 11 "2/28 ARM" WL 00 WAC 7.5443 (
42625095.51 / 42625095.51 ); 42625095.51
0.5032 0.5032 358:2 358:2 360 NO_CHECK ARM
LIBOR_6MO 6.407 23 6 SYNC_INT 13.5443
1 7.5443 0 0 INIT_PERCAP 2 GROUP "11A"
TEASER
M 12 "2/28 ARM" WL 00 WAC 7.5345 (
130302907.14 / 130302907.14 ); 130302907.14

```

0.5032	0.5032	358:2	358:2	360	NO_CHECK ARM
LIBOR_6MO		6.3594	23	6	SYNC_INT 13.5345
1	7.5345	0	0	INIT_PERCAP	2 GROUP "11A"
TEASER					
M	13	"2/28 ARM"	WL	00	WAC 7.0564 (
27805752.31	/	27805752.31	);	27805752.31	
0.5032	0.5032	359:1	359:1	360	NO_CHECK ARM
LIBOR_6MO		6.3584	24	6	SYNC_INT 13.0564
1	7.0564	0	0	INIT_PERCAP	2 GROUP "11A"
TEASER					
M	14	"3/17 ARM"	WL	00	WAC 7.55 (
159712.82	/	159712.82	);	159712.82	
0.5032	0.5032	239:1	239:1	240	NO_CHECK ARM
LIBOR_6MO		6.5	36	6	SYNC_INT 13.55
1	7.55	0	0	INIT_PERCAP	2 GROUP "11A"
TEASER					
M	15	"3/27 ARM"	WL	00	WAC 7.3067 (
43553757.86	/	43553757.86	);	43553757.86	
0.5032	0.5032	359:1	359:1	360	NO_CHECK ARM
LIBOR_6MO		6.3416	36	6	SYNC_INT 13.3067
1	7.3067	0	0	INIT_PERCAP	2 GROUP "11A"
TEASER					
M	16	"3/27 ARM"	WL	00	WAC 7.5641 (
1872007.01	/	1872007.01	);	1872007.01	
0.5032	0.5032	359:1	359:1	360	NO_CHECK ARM
LIBOR_6MO		6.5	36	6	SYNC_INT 13.5641
1	7.5641	0	0	INIT_PERCAP	2 GROUP "11A"
TEASER					
M	17	"3/27 ARM"	WL	00	WAC 7.1661 (
27610641.72	/	27610641.72	);	27610641.72	
0.5032	0.5032	359:1	359:1	360	NO_CHECK ARM
LIBOR_6MO		6.3406	36	6	SYNC_INT 13.1661
1	7.1661	0	0	INIT_PERCAP	2 GROUP "11A"
TEASER					
M	18	"15 YR FIXED"	WL	00	WAC 7.5417 (
494402.52	/	494402.52	);	494402.52	
0.5032	0.5032	178:2	178:2	180	NO_CHECK
GROUP "22F"					
M	19	"15 YR FIXED"	WL	00	WAC 7.4548 (
768981.18	/	768981.18	);	768981.18	
0.5032	0.5032	176:4	176:4	180	NO_CHECK
GROUP "22F"					
M	20	"15 YR FIXED"	WL	00	WAC 7.0649 (
1974847.34	/	1974847.34	);	1974847.34	
0.5032	0.5032	178:2	178:2	180	NO_CHECK
GROUP "22F"					
M	21	"20 YR FIXED"	WL	00	WAC 7.2418 (
496232.51	/	496232.51	);	496232.51	
0.5032	0.5032	237:3	237:3	240	NO_CHECK
GROUP "22F"					
M	22	"20 YR FIXED"	WL	00	WAC 7.9 (
64688.61	/	64688.61	);	64688.61	
0.5032	0.5032	239:1	239:1	240	NO_CHECK
GROUP "22F"					
M	23	"20 YR FIXED"	WL	00	WAC 6.5527 (
1115240.00	/	1115240.00	);	1115240.00	
0.5032	0.5032	238:2	238:2	240	NO_CHECK
GROUP "22F"					
M	24	"30 YR FIXED"	WL	00	WAC 7.7501 (
11376436.13	/	11376436.13	);	11376436.13	
0.5032	0.5032	358:2	358:2	360	NO_CHECK

GROUP "22F"

M	25	"30 YR FIXED"	WL	00	WAC	7.1394 (
11460721.45 /		11460721.45);			11460721.45	
0.5032	0.5032	357:3		357:3	360 NO_CHECK	

GROUP "22F"

M	26	"30 YR FIXED"	WL	00	WAC	6.9086 (
49589880.55 /		49589880.55);			49589880.55	
0.5032	0.5032	358:2		358:2	360 NO_CHECK	

GROUP "22F"

M	27	"2/13 ARM"	WL	00	WAC	8.65 (
67814.55 /		67814.55);			67814.55	
0.5032	0.5032	179:1		179:1	180 NO_CHECK ARM	
LIBOR_6MO		6.5	24	6 SYNC_INT	14.65	
1	8.65	0	0	INIT_PERCAP	2 GROUP "22A"	

TEASER

M	28	"2/13 ARM"	WL	00	WAC	6.95 (
80494.13 /		80494.13);			80494.13	
0.5032	0.5032	179:1		179:1	180 NO_CHECK ARM	
LIBOR_6MO		6.5	24	6 SYNC_INT	12.95	
1	6.95	0	0	INIT_PERCAP	2 GROUP "22A"	

TEASER

M	29	"2/18 ARM"	WL	00	WAC	6.9966 (
144722.19 /		144722.19);			144722.19	
0.5032	0.5032	239:1		239:1	240 NO_CHECK ARM	
LIBOR_6MO		6.5	24	6 SYNC_INT	12.9966	
1	6.9966	0	0	INIT_PERCAP	2 GROUP "22A"	

TEASER

M	30	"2/28 ARM"	WL	00	WAC	7.8605 (
14283184.35 /		14283184.35);			14283184.35	
0.5032	0.5032	358:2		358:2	360 NO_CHECK ARM	
LIBOR_6MO		6.357	23	6 SYNC_INT	13.8605	
1	7.8605	0	0	INIT_PERCAP	2 GROUP "22A"	

TEASER

M	31	"2/28 ARM"	WL	00	WAC	7.4459 (
66122490.90 /		66122490.90);			66122490.90	
0.5032	0.5032	358:2		358:2	360 NO_CHECK ARM	
LIBOR_6MO		6.3736	23	6 SYNC_INT	13.4459	
1	7.4459	0	0	INIT_PERCAP	2 GROUP "22A"	

TEASER

M	32	"2/28 ARM"	WL	00	WAC	7.1653 (
12784611.70 /		12784611.70);			12784611.70	
0.5032	0.5032	359:1		359:1	360 NO_CHECK ARM	
LIBOR_6MO		6.3614	24	6 SYNC_INT	13.1653	
1	7.1653	0	0	INIT_PERCAP	2 GROUP "22A"	

TEASER

M	33	"3/27 ARM"	WL	00	WAC	7.436 (
14769957.70 /		14769957.70);			14769957.70	
0.5032	0.5032	359:1		359:1	360 NO_CHECK ARM	
LIBOR_6MO		6.3903	36	6 SYNC_INT	13.436	
1	7.436	0	0	INIT_PERCAP	2 GROUP "22A"	

TEASER

M	34	"3/27 ARM"	WL	00	WAC	7.3458 (
1131467.30 /		1131467.30);			1131467.30	
0.5032	0.5032	359:1		359:1	360 NO_CHECK ARM	
LIBOR_6MO		5.896	36	6 SYNC_INT	13.3458	
1	7.3458	0	0	INIT_PERCAP	2 GROUP "22A"	

TEASER

M	35	"3/27 ARM"	WL	00	WAC	7.1722 (
10149356.74 /		10149356.74);			10149356.74	
0.5032	0.5032	359:1		359:1	360 NO_CHECK ARM	
LIBOR_6MO		6.3615	36	6 SYNC_INT	13.1722	

1	7.1722	0	0	INIT_PERCAP	2	GROUP "22A"
TEASER						
M	36	"15 YR FIXED"	WL	00	WAC	6.0435 (
262709.07 /		262709.07);		262709.07		
0.5032	0.5032	180:0		180:0	180	NO_CHECK
GROUP "11F"		PREFUND 1 at 0.5032				
M	37	"15 YR FIXED"	WL	00	WAC	7.0271 (
1025712.19 /		1025712.19);		1025712.19		
0.5032	0.5032	180:0		180:0	180	NO_CHECK
GROUP "11F"		PREFUND 1 at 0.5032				
M	38	"20 YR FIXED"	WL	00	WAC	6.4632 (
65767.88 /		65767.88);		65767.88		
0.5032	0.5032	240:0		240:0	240	NO_CHECK
GROUP "11F"		PREFUND 1 at 0.5032				
M	39	"20 YR FIXED"	WL	00	WAC	6.8349 (
209719.98 /		209719.98);		209719.98		
0.5032	0.5032	240:0		240:0	240	NO_CHECK
GROUP "11F"		PREFUND 1 at 0.5032				
M	40	"20 YR FIXED"	WL	00	WAC	7.1613 (
616967.54 /		616967.54);		616967.54		
0.5032	0.5032	240:0		240:0	240	NO_CHECK
GROUP "11F"		PREFUND 1 at 0.5032				
M	41	"30 YR FIXED"	WL	00	WAC	7.5813 (
5579396.54 /		5579396.54);		5579396.54		
0.5032	0.5032	360:0		360:0	360	NO_CHECK
GROUP "11F"		PREFUND 1 at 0.5032				
M	42	"30 YR FIXED"	WL	00	WAC	7.1019 (
5456905.57 /		5456905.57);		5456905.57		
0.5032	0.5032	360:0		360:0	360	NO_CHECK
GROUP "11F"		PREFUND 1 at 0.5032				
M	43	"30 YR FIXED"	WL	00	WAC	7.0514 (
17252052.86 /		17252052.86);		17252052.86		
0.5032	0.5032	360:0		360:0	360	NO_CHECK
GROUP "11F"		PREFUND 1 at 0.5032				
M	44	"2/18 ARM"	WL	00	WAC	6.15 (
56093.16 /		56093.16);		56093.16		
0.5032	0.5032	240:0		240:0	240	NO_CHECK ARM
LIBOR_6MO		6.5		26	6 SYNC_INT	12.15
1	6.15	0	0	INIT_PERCAP	2	GROUP "11A"
PREFUND 1 at	0.5032	TEASER				
M	45	"2/18 ARM"	WL	00	WAC	5.95 (
40204.92 /		40204.92);		40204.92		
0.5032	0.5032	240:0		240:0	240	NO_CHECK ARM
LIBOR_6MO		6.5		26	6 SYNC_INT	11.95
1	5.95	0	0	INIT_PERCAP	2	GROUP "11A"
PREFUND 1 at	0.5032	TEASER				
M	46	"2/28 ARM"	WL	00	WAC	7.5443 (
14208361.29 /		14208361.29);		14208361.29		
0.5032	0.5032	360:0		360:0	360	NO_CHECK ARM
LIBOR_6MO		6.407		26	6 SYNC_INT	13.5443
1	7.5443	0	0	INIT_PERCAP	2	GROUP "11A"
PREFUND 1 at	0.5032	TEASER				
M	47	"2/28 ARM"	WL	00	WAC	7.5345 (
43434290.52 /		43434290.52);		43434290.52		
0.5032	0.5032	360:0		360:0	360	NO_CHECK ARM
LIBOR_6MO		6.3594		26	6 SYNC_INT	13.5345
1	7.5345	0	0	INIT_PERCAP	2	GROUP "11A"
PREFUND 1 at	0.5032	TEASER				
M	48	"2/28 ARM"	WL	00	WAC	7.0564 (
9268581.57 /		9268581.57);		9268581.57		
0.5032	0.5032	360:0		360:0	360	NO_CHECK ARM

LIBOR_6MO		6.3584	26	6 SYNC_INT	13.0564
1	7.0564	0	0	INIT_PERCAP	2 GROUP "11A"
PREFUND 1 at	0.5032	TEASER			
M	49	"3/17 ARM"	WL	00 WAC	7.55 (
53237.60 /	53237.60	);		53237.60	
0.5032	0.5032	240:0	240:0	240	NO_CHECK ARM
LIBOR_6MO		6.5	38	6 SYNC_INT	13.55
1	7.55	0	0	INIT_PERCAP	2 GROUP "11A"
PREFUND 1 at	0.5032	TEASER			
M	50	"3/27 ARM"	WL	00 WAC	7.3067 (
14517915.32 /	14517915.32	);		14517915.32	
0.5032	0.5032	360:0	360:0	360	NO_CHECK ARM
LIBOR_6MO		6.3416	38	6 SYNC_INT	13.3067
1	7.3067	0	0	INIT_PERCAP	2 GROUP "11A"
PREFUND 1 at	0.5032	TEASER			
M	51	"3/27 ARM"	WL	00 WAC	7.5641 (
624002.17 /	624002.17	);		624002.17	
0.5032	0.5032	360:0	360:0	360	NO_CHECK ARM
LIBOR_6MO		6.5	38	6 SYNC_INT	13.5641
1	7.5641	0	0	INIT_PERCAP	2 GROUP "11A"
PREFUND 1 at	0.5032	TEASER			
M	52	"3/27 ARM"	WL	00 WAC	7.1661 (
9203544.73 /	9203544.73	);		9203544.73	
0.5032	0.5032	360:0	360:0	360	NO_CHECK ARM
LIBOR_6MO		6.3406	38	6 SYNC_INT	13.1661
1	7.1661	0	0	INIT_PERCAP	2 GROUP "11A"
PREFUND 1 at	0.5032	TEASER			
M	53	"15 YR FIXED"	WL	00 WAC	7.5417 (
164724.72 /	164724.72	);		164724.72	
0.5032	0.5032	180:0	180:0	180	NO_CHECK
GROUP "22F"	PREFUND 1 at	0.5032			
M	54	"15 YR FIXED"	WL	00 WAC	7.4548 (
256208.67 /	256208.67	);		256208.67	
0.5032	0.5032	180:0	180:0	180	NO_CHECK
GROUP "22F"	PREFUND 1 at	0.5032			
M	55	"15 YR FIXED"	WL	00 WAC	7.0649 (
657978.41 /	657978.41	);		657978.41	
0.5032	0.5032	180:0	180:0	180	NO_CHECK
GROUP "22F"	PREFUND 1 at	0.5032			
M	56	"20 YR FIXED"	WL	00 WAC	7.2418 (
165334.44 /	165334.44	);		165334.44	
0.5032	0.5032	240:0	240:0	240	NO_CHECK
GROUP "22F"	PREFUND 1 at	0.5032			
M	57	"20 YR FIXED"	WL	00 WAC	7.9 (
21552.91 /	21552.91	);		21552.91	
0.5032	0.5032	240:0	240:0	240	NO_CHECK
GROUP "22F"	PREFUND 1 at	0.5032			
M	58	"20 YR FIXED"	WL	00 WAC	6.5527 (
371574.97 /	371574.97	);		371574.97	
0.5032	0.5032	240:0	240:0	240	NO_CHECK
GROUP "22F"	PREFUND 1 at	0.5032			
M	59	"30 YR FIXED"	WL	00 WAC	7.7501 (
3790393.92 /	3790393.92	);		3790393.92	
0.5032	0.5032	360:0	360:0	360	NO_CHECK
GROUP "22F"	PREFUND 1 at	0.5032			
M	60	"30 YR FIXED"	WL	00 WAC	7.1394 (
3818476.05 /	3818476.05	);		3818476.05	
0.5032	0.5032	360:0	360:0	360	NO_CHECK
GROUP "22F"	PREFUND 1 at	0.5032			
M	61	"30 YR FIXED"	WL	00 WAC	6.9086 (
16522325.62 /	16522325.62	);		16522325.62	

0.5032	0.5032	360:0	360:0	360	NO_CHECK	
GROUP "22F" PREFUND 1 at 0.5032						
M	62	"2/13 ARM"	WL	00	WAC	8.65 (
22611.20 /		22611.20);		22611.20		
0.5032	0.5032	180:0	180:0	180	NO_CHECK ARM	
LIBOR_6MO		6.5	26	6	SYNC_INT	14.65
1	8.65	0	0	INIT_PERCAP	2	GROUP "22A"
PREFUND 1 at 0.5032 TEASER						
M	63	"2/13 ARM"	WL	00	WAC	6.95 (
26838.93 /		26838.93);		26838.93		
0.5032	0.5032	180:0	180:0	180	NO_CHECK ARM	
LIBOR_6MO		6.5	26	6	SYNC_INT	12.95
1	6.95	0	0	INIT_PERCAP	2	GROUP "22A"
PREFUND 1 at 0.5032 TEASER						
M	64	"2/18 ARM"	WL	00	WAC	6.9966 (
48254.29 /		48254.29);		48254.29		
0.5032	0.5032	240:0	240:0	240	NO_CHECK ARM	
LIBOR_6MO		6.5	26	6	SYNC_INT	12.9966
1	6.9966	0	0	INIT_PERCAP	2	GROUP "22A"
PREFUND 1 at 0.5032 TEASER						
M	65	"2/28 ARM"	WL	00	WAC	7.8605 (
4762399.81 /		4762399.81);		4762399.81		
0.5032	0.5032	360:0	360:0	360	NO_CHECK ARM	
LIBOR_6MO		6.357	26	6	SYNC_INT	13.8605
1	7.8605	0	0	INIT_PERCAP	2	GROUP "22A"
PREFUND 1 at 0.5032 TEASER						
M	66	"2/28 ARM"	WL	00	WAC	7.4459 (
22047026.12 /		22047026.12);		22047026.12		
0.5032	0.5032	360:0	360:0	360	NO_CHECK ARM	
LIBOR_6MO		6.3736	26	6	SYNC_INT	13.4459
1	7.4459	0	0	INIT_PERCAP	2	GROUP "22A"
PREFUND 1 at 0.5032 TEASER						
M	67	"2/28 ARM"	WL	00	WAC	7.1653 (
4262735.18 /		4262735.18);		4262735.18		
0.5032	0.5032	360:0	360:0	360	NO_CHECK ARM	
LIBOR_6MO		6.3614	26	6	SYNC_INT	13.1653
1	7.1653	0	0	INIT_PERCAP	2	GROUP "22A"
PREFUND 1 at 0.5032 TEASER						
M	68	"3/27 ARM"	WL	00	WAC	7.436 (
4924703.19 /		4924703.19);		4924703.19		
0.5032	0.5032	360:0	360:0	360	NO_CHECK ARM	
LIBOR_6MO		6.3903	38	6	SYNC_INT	13.436
1	7.436	0	0	INIT_PERCAP	2	GROUP "22A"
PREFUND 1 at 0.5032 TEASER						
M	69	"3/27 ARM"	WL	00	WAC	7.3458 (
377261.79 /		377261.79);		377261.79		
0.5032	0.5032	360:0	360:0	360	NO_CHECK ARM	
LIBOR_6MO		5.896	38	6	SYNC_INT	13.3458
1	7.3458	0	0	INIT_PERCAP	2	GROUP "22A"
PREFUND 1 at 0.5032 TEASER						
M	70	"3/27 ARM"	WL	00	WAC	7.1722 (
3384069.93 /		3384069.93);		3384069.93		
0.5032	0.5032	360:0	360:0	360	NO_CHECK ARM	
LIBOR_6MO		6.3615	38	6	SYNC_INT	13.1722
1	7.1722	0	0	INIT_PERCAP	2	GROUP "22A"
PREFUND 1 at 0.5032 TEASER						