SECURITIES AND EXCHANGE COMMISSION
WASHINGTON, DC 20549

RECD S.E.C.
JUL 20 2004
1086

FORM SE

FORM FOR SUBMISSION OF PAPER FORMAT EXHIBITS
BY ELECTRONIC FILERS

Saxon Asset Securities Company
(Exact Name of Registrant as Specified in Charter)

0001014299
(Registrant CIK Number)

Form 8-K for July 19, 2004
(Electronic Report, Schedule or Registration Statement of Which the Documents Are a Part (Give Period of Report))

333-111832
(SEC File Number, if Available)

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

PROCESSED
JUL 23 2004
THOMSON
FINANCIAL

SIGNATURES

Filings Made by the Registrant. The registrant has duly caused this form to be signed on its behalf by the undersigned, thereunto duly authorized, in the state of Virginia, on July 19, 2004.

SAXON ASSET SECURITIES COMPANY

By: _____

Name: Robert B. Eastep
Title: Executive Vice President and Chief Financial Officer

Exhibit Index

Exhibit		Page
99.1	Computational Materials	5
99.2	Computational Materials	6
99.3	Computational Materials	7
99.4	Computational Materials	8
99.5	Computational Materials	9
99.6	Computational Materials	10

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

COMPUTATIONAL MATERIALS

for

SAXON ASSET SECURITIES COMPANY

Saxon Asset Securities Trust 2004-2
Mortgage Loan Asset Backed Securities, Series 2004-2

Saxon Asset Securities Trust 2004-2 - Stack Price/Yield

Settle	7/27/2004
First Payment	8/25/2004

Assumptions
100% Prepay
CDR Break
40% Severity
12 mo lag
100% Advances
Triggers Fail

	FLAT LIBOR		FWD LIBOR	
	CDR BREAK	CUM LOSS	CDR BREAK	CUM LOSS
MF1	12.2 CDR	63,934,717.16 (13.99%)	11.3 CDR	60,492,524.26 (13.24%)
MF2	8.8 CDR	50,097,757.64 (10.96%)	7.6 CDR	44,621,495.65 (9.77%)
MF3	8.5 CDR	48,760,373.60 (10.67%)	6.5 CDR	39,289,112.64 (8.60%)
MF4	8.2 CDR	47,402,188.10 (10.37%)	5.9 CDR	36,244,833.16 (7.93%)
MF5	7.8 CDR	45,558,091.95 (9.97%)	5.4 CDR	33,630,431.67 (7.36%)
MF6	7.5 CDR	44,149,495.24 (9.66%)	4.7 CDR	29,846,089.26 (6.53%)
MV1	23.3 CDR	133,278,824.24 (17.94%)	19.7 CDR	118,672,281.25 (15.97%)
MV2	15.3 CDR	98,388,343.47 (13.24%)	12.0 CDR	81,472,652.39 (10.96%)
MV3	13.2 CDR	87,761,644.41 (11.81%)	10 CDR	70,227,094.98 (9.45%)
MV4	11.3 CDR	77,512,987.01 (10.43%)	8.2 CDR	59,429,917.47 (8.00%)
MV5	10.1 CDR	70,701,387.32 (9.51%)	7.2 CDR	53,129,622.98 (7.15%)
MV6	9.0 CDR	64,209,572.98 (8.64%)	6.3 CDR	47,261,866.12 (6.36%)

Saxon Asset Securities Trust 2004-2 - Stack Price/Yield

Settle	7/27/2004
First Payment	8/25/2004

	0	50	75	100	125	150	175
AF2							
Price	100	100	100	100	100	100	100
WAL	19.89	5.37	3.60	2.70	2.17	1.82	1.57
Mod Durn	13.35	4.72	3.28	2.51	2.04	1.72	1.49
Principal Window Begin	213	50	34	26	21	18	16
Principal Window End	266	88	56	41	31	26	22
LIBOR_1MO	1.367	1.367	1.367	1.367	1.367	1.367	1.367
LIBOR_6MO	1.840	1.840	1.840	1.840	1.840	1.84	1.84
CMT_1YR	1.990	1.990	1.990	1.990	1.990	1.99	1.99
Prepay	0 PricingSpeed	50 PricingSpeed	75 PricingSpeed	100 PricingSpeed	125 PricingSpeed	150 PricingSpeed	175 PricingSpeed
No Prepays							
Lockout and Penalties	Include Penalties	Include Penalties	Include Penalties	Include Penalties	Include Penalties	Include Penalties	Include Penalties
Prepay Penalty Haircut	5	5	5	5	5	5	5
Optional Redemption	Call (Y,Y)	Call (Y,Y)	Call (Y,Y)	Call (Y,Y)	Call (Y,Y)	Call (Y,Y)	Call (Y,Y)

Saxon Asset Securities Trust 2004-2 - Stack Price/Yield

Settle	7/27/2004
First Payment	8/25/2004

	0	50	75	100	125	150	175
AF2							
Price	100	100	100	100	100	100	100
WAL	19.89	5.37	3.60	2.70	2.17	1.82	1.57
Mod Durn	13.35	4.72	3.28	2.51	2.04	1.72	1.49
Principal Window Begin	213	50	34	26	21	18	16
Principal Window End	266	88	56	41	31	26	22
LIBOR_1MO	1.367	1.367	1.367	1.367	1.367	1.367	1.367
LIBOR_6MO	1.840	1.840	1.840	1.840	1.840	1.84	1.84
CMT_1YR	1.990	1.990	1.990	1.990	1.990	1.99	1.99
Prepay	0 PricingSpeed	50 PricingSpeed	75 PricingSpeed	100 PricingSpeed	125 PricingSpeed	150 PricingSpeed	175 PricingSpeed
No Prepays							
Lockout and Penalties	Include Penalties	Include Penalties	Include Penalties	Include Penalties	Include Penalties	Include Penalties	Include Penalties
Prepay Penalty Haircut	5	5	5	5	5	5	5
Optional Redemption	Call (N,N)	Call (N,N)	Call (N,N)	Call (N,N)	Call (N,N)	Call (N,N)	Call (N,N)

Saxon 2004-2
FOR INTERNAL DISTIBUTION ONLY

Prepayment Speed		50% Pricing	50% Pricing	50% Pricing	50% Pricing	50% Pricing	50% Pricing	50% Pricing	50% Pricing	50% Pricing
Losses		CDR	CDR	CDR	CDR	CDR	CDR	CDR	CDR	CDR
Loss Severity		40%	40%	40%	50%	50%	50%	60%	60%	60%
Lag		12	12	12	12	12	12	12	12	12
LIBOR		Static	FWD	FWD + 200	Static	FWD	FWD + 200	Static	FWD	FWD + 200
Triggers		Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail
Optional Redemption		To Maturity	To Maturity	To Maturity	To Maturity	To Maturity	To Maturity	To Maturity	To Maturity	To Maturity
Class MV1										
WAL	0	9.51	10.7	12.43	10.82	12.01	13.55	11.79	12.92	14.28
Cum Loss		22.39%	19.77%	15.84%	23.89%	20.76%	16.40%	25.00%	21.47%	16.77%
Cum Default		55.46%	49.03%	39.27%	47.30%	41.18%	32.52%	41.22%	35.46%	27.71%
CDR Multiple		19.14 CDR	15.17 CDR	10.59 CDR	14.43 CDR	11.42 CDR	8.07 CDR	11.58 CDR	9.15 CDR	6.51 CDR
Class MV2										
WAL		11.82	13.35	15.31	12.97	14.39	16.11	13.8	15.08	16.62
Cum Loss		18.46%	15.36%	10.97%	19.64%	16.07%	11.32%	20.51%	16.56%	11.55%
Cum Default		45.68%	38.06%	27.18%	38.86%	31.84%	22.43%	33.79%	27.33%	19.06%
CDR Multiple		13.62 CDR	10.14 CDR	6.35 CDR	10.59 CDR	7.87 CDR	4.98 CDR	8.66 CDR	6.42 CDR	4.09 CDR
Class MV3										
WAL		16.02	17.98	20.4	17.27	19.13	21.06	18.22	19.97	21.59
Cum Loss		17.29%	14.03%	9.58%	18.37%	14.66%	9.86%	19.17%	15.12%	10.07%
Cum Default		42.77%	34.75%	23.72%	36.34%	29.03%	19.53%	31.58%	24.94%	16.61%
CDR Multiple		12.26 CDR	8.89 CDR	5.34 CDR	9.6 CDR	6.95 CDR	4.21 CDR	7.89 CDR	5.71 CDR	3.48 CDR

This information is being provided in response to your specific request for information. The information has been prepared and furnished to you solely by CREDIT SUISSE FIRST BOSTON CORPORATION (CSFBC) and not the Issuer of the Securities or any of its affiliates. The preliminary description of the underlying assests has not been independently verified by CSFBC. CSFBC is not acting as agent for the Issuer or its affiliiates in connection with the proposed transaction. All information contained herein is preliminary, limited in nature and subject to completion or amendment. CSFBC makes no representation that the above referenced seurity will actually perform as described in any scenario. The above analysis alone is not intended to be a prospectus and any investment decision with respect to the security should be made by you based solely upon the information contained in the final prospectus. Under no circumstances shall the information presented constitute an offer to sell or solicitation of an offer to by nor shall there be any sale of securities in any jurisdiction in which such offer, solicitation or sale would be unlawful prior to registration or qualification under the securities laws of such jurisdiction. The securities my not be sold nor may an offer to buy be accepted prior to the delivery of a final prospectus relating to the securites.

Saxon 2004-2
FOR INTERNAL DISTIBUTION ONLY

Prepayment Speed		100% Pricing	100% Pricing	100% Pricing	100% Pricing	100% Pricing	100% Pricing	100% Pricing	100% Pricing	100% Pricing
Losses		CDR	CDR	CDR	CDR	CDR	CDR	CDR	CDR	CDR
Loss Severity		40%	40%	40%	50%	50%	50%	60%	60%	60%
Lag		12	12	12	12	12	12	12	12	12
LIBOR		Static	FWD	FWD + 200	Static	FWD	FWD + 200	Static	FWD	FWD + 200
Triggers		Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail
Optional Redemption		To Maturity	To Maturity	To Maturity	To Maturity	To Maturity	To Maturity	To Maturity	To Maturity	To Maturity
Class MV1										
WAL	0	5.96	6.33	6.97	6.54	6.85	7.43	6.93	7.22	7.73
Cum Loss		17.95%	15.98%	12.81%	18.50%	16.29%	12.96%	18.87%	16.50%	13.06%
Cum Default		44.54%	39.67%	31.81%	36.71%	32.35%	25.74%	31.20%	27.31%	21.62%
CDR Multiple		23.32 CDR	19.71 CDR	14.62 CDR	17.75 CDR	14.96 CDR	11.19 CDR	14.31 CDR	12.05 CDR	9.06 CDR
Class MV2										
WAL		7.33	7.78	8.57	7.79	8.19	8.9	8.12	8.46	9.08
Cum Loss		13.25%	10.99%	7.62%	13.62%	11.17%	7.71%	13.88%	11.29%	7.75%
Cum Default		32.86%	27.27%	18.91%	27.03%	22.18%	15.31%	22.95%	18.68%	12.82%
CDR Multiple		15.31 CDR	12.03 CDR	7.74 CDR	11.93 CDR	9.35 CDR	6.08 CDR	9.77 CDR	7.64 CDR	4.99 CDR
Class MV3										
WAL		10	10.7	11.84	10.62	11.18	12.13	11.07	11.53	12.41
Cum Loss		11.87%	9.52%	6.13%	12.20%	9.67%	6.19%	12.44%	9.78%	6.23%
Cum Default		29.43%	23.63%	15.22%	24.21%	19.20%	12.29%	20.55%	16.18%	10.31%
CDR Multiple		13.28 CDR	10.09 CDR	6.04 CDR	10.42 CDR	7.89 CDR	4.76 CDR	8.57 CDR	6.48 CDR	3.93 CDR

This information is being provided in response to your specific request for information. The information has been prepared and furnished to you solely by CREDIT SUISSE FIRST BOSTON CORPORATION (CSFBC) and not the Issuer of the Securities or any of its affiliates. The preliminary description of the underlying assests has not been independently verified by CSFBC. CSFBC is not acting as agent for the Issuer or its affiliiates in connection with the proposed transaction. All information contained herein is preliminary, limited in nature and subject to completion or amendment. CSFBC makes no representation that the above referenced seurity will actually perform as described in any scenario. The above analysis alone is not intended to be a prospectus and any investment decision with respect to the security should be made by you based solely upon the information contained in the final prospectus. Under no circumstances shall the information presented constitute an offer to sell or solicitation of an offer to by nor shall there be any sale of securities in any jurisdiction in which such offer, solicitation or sale would be unlawful prior to registration or qualification under the securities laws of such jurisdiction. The securities my not be sold nor may an offer to buy be accepted prior to the delivery of a final prospectus relating to the securites.

```
! SAST0402.CDI #CMOVER_3.0D ASSET_BACKED_HOMEEQUITY ! MAX_CF_VECTSIZE 620
!
!! Created by Intex Deal Maker v3.7.084 , subroutines 3.1
!! 07/14/2004 5:40 AM
!
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 give 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. "
!
! Modeled in the Intex CMO Modeling Language, (WNYC1255538)
! 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"
 GROUP 2 = "2C_A" "2C_F" "2NC_A" "2NC_F"
!
  DEFINE PREPAY PPC  GROUP "1F" RISE_PERS 10 START_CPR 2.2 END_CPR 22
  DEFINE PREPAY PPC GROUP "2C_A" RAMP 8 10 12 14 16 18 20 22 24 26 28 30 30 30 30 30 30 30
30 30 30 30 55 55 55 55 55 35
!
  DEFINE PREPAY PPC  GROUP "2C_F" RISE_PERS 10 START_CPR 2.2 END_CPR 22
  DEFINE PREPAY PPC GROUP "2NC_A" RAMP 8 10 12 14 16 18 20 22 24 26 28 30 30 30 30 30 30 30
30 30 30 30 55 55 55 55 55 35
!
  DEFINE PREPAY PPC  GROUP "2NC_F" RISE_PERS 10 START_CPR 2.2 END_CPR 22
!
  DEFINE CONSTANT #OrigCollBal = 1200000000.00
  DEFINE CONSTANT #OrigCollBal1 = 456930761.12
  DEFINE CONSTANT #OrigCollBal2 = 743069238.88
  DEFINE CONSTANT #OrigCollBal2C_A = 315863412.04
  DEFINE CONSTANT #OrigCollBal2C_F = 55669465.51
  DEFINE CONSTANT #OrigCollBal2NC_A = 316024228.82
  DEFINE CONSTANT #OrigCollBal2NC_F = 55512132.51
!
```

```
 DEFINE CONSTANT #OrigBondBal = 1200000000.00
 DEFINE CONSTANT #OrigBondBal1 = 456930000.00
 DEFINE CONSTANT #OrigBondBal2 = 743069000.00
!
 DEFINE CONSTANT #SpecSenEnhPct1 = 32%
 DEFINE CONSTANT #AFTargPct = 68.000000000000%
 DEFINE CONSTANT #MF1TargPct = 78.000000000000%
 DEFINE CONSTANT #MF2TargPct = 85.500000000000%
 DEFINE CONSTANT #MF3TargPct = 88.000000000000%
 DEFINE CONSTANT #MF4TargPct = 90.000000000000%
 DEFINE CONSTANT #MF5TargPct = 92.000000000000%
 DEFINE CONSTANT #MF6TargPct = 94.000000000000%
 DEFINE CONSTANT #SpecSenEnhPct2 = 38.9%
 DEFINE CONSTANT #AVTargPct = 61.100000000000%
 DEFINE CONSTANT #MV1TargPct = 74.100000000000%
 DEFINE CONSTANT #MV2TargPct = 84.600000000000%
 DEFINE CONSTANT #MV3TargPct = 87.600000000000%
 DEFINE CONSTANT #MV4TargPct = 90.600000000000%
 DEFINE CONSTANT #MV5TargPct = 92.600000000000%
 DEFINE CONSTANT #MV6TargPct = 94.600000000000%
 DEFINE #BondBal1              = 456930000.00
 DEFINE #BondBal2              = 743069000.00
 DEFINE #BondBal               = 1199999000.00
!
  FULL_DEALNAME:       Saxon Asset Securities Trust 2004-2
  TRUSTEE_DEAL:        Deutsche Bank Trust Company Americas
  TRUSTEE_INDENTURE:   Deutsche Bank Trust Company Americas
  TRUSTEE_OWNER:       Wilmington Trust Company
  SERVICER_MASTER:     Saxon Mortgage Services
!
  ISSUER:            Saxon Asset Securities Trust 2004-2
  DEALER:            Credit Suisse First Boston
  DEAL SIZE:         $ 1199999000.00
  PRICING SPEED:       GROUP "1F" HEP 22%
  PRICING SPEED:       GROUP "2C_A" PPC 100%
  PRICING SPEED:       GROUP "2C_F" HEP 22%
  PRICING SPEED:       GROUP "2NC_A" PPC 100%
  PRICING SPEED:       GROUP "2NC_F" HEP 22%
! ISSUE DATE:         20040701
  SETTLEMENT DATE:     20040727
!
 Record date delay:    24
!
 DEFINE TR_INDEXDEPS_ALL
!
DEFINE TRANCHE "CAP_IN12", "CAP_IN22", "CAP_IN32", "AIOC"[2], "AIONC"[2], "AF5", "AF1", "AF2",
"AF3", "AF4", "MF1", "MF2", "MF3", "MF4", "MF5", "MF6", "AV1", "AV2", "MV1", "MV2", "MV3", "MV4",
"MV5", "MV6", "R_1", "R_PP_1", "R_2", "R_PP_2"
!
 DEFINE SCHEDULE
"AIOCA","AIOCF","AIONCA","AIONCF","AIOCA_BAL","AIOCF_BAL","AIONCA_BAL","AIONCF_BAL"
!
 DEFINE VARNAMES #Coll_I1, #Coll_P1, #Coll_I2, #Coll_P2, #TrigDelinqFrac1, #ReqPerc1,
#TrigEnhFrac1, #CumLossShft1, #TrigCumLossFrac1, #TrigDelinqFrac2, #ReqPerc2, #TrigEnhFrac2,
#CumLossShft2, #TrigCumLossFrac2
```

```
!
  DEAL_CLOCK_INFO _
    ISSUE_CDU_DATE          20040701 _
    DEAL_FIRSTPAY_DATE       20040825
!
!
 DEFINE TABLE "Cap12Notional" (53, 2) = "CURDATE" "Balance"
    20040825.1    0
    20040925.1    0
    20041025.1    728,713,221
    20041125.1    719,609,091
    20041225.1    709,285,017
    20050125.1    697,763,926
    20050225.1    685,075,168
    20050325.1    671,251,174
    20050425.1    656,338,055
    20050525.1    640,463,520
    20050625.1    623,748,901
    20050725.1    606,446,892
    20050825.1    589,357,098
    20050925.1    572,520,576
    20051025.1    555,936,171
    20051125.1    539,828,674
    20051225.1    524,183,901
    20060125.1    508,988,572
    20060225.1    494,217,757
    20060325.1    479,855,792
    20060425.1    465,832,613
    20060525.1    450,032,727
    20060625.1    426,478,141
    20060725.1    404,310,158
    20060825.1    383,563,532
    20060925.1    361,566,698
    20061025.1    342,197,571
    20061125.1    328,934,638
    20061225.1    316,250,786
    20070125.1    304,107,503
    20070225.1    293,946,317
    20070325.1    284,126,888
    20070425.1    274,637,898
    20070525.1    265,467,135
    20070625.1    256,604,273
    20070725.1    248,060,025
    20070825.1    239,817,217
    20070925.1    231,850,532
    20071025.1    224,150,533
    20071125.1    216,714,420
    20071225.1    209,525,941
    20080125.1    202,579,118
    20080225.1    195,866,230
    20080325.1    189,376,715
    20080425.1    183,103,068
    20080525.1    177,037,302
    20080625.1    171,172,395
    20080725.1    165,502,663
```

```
    20080825.1    160,021,198
    20080925.1    154,720,971
    20081025.1    149,595,735
    20081125.1    144,639,439
    20081225.1    0
!
DEFINE TABLE "Cap12RateSch" (53, 2) = "CURDATE" "Rate"
    20040825.1    0
    20040925.1    0
    20041025.1    5.21
    20041125.1    5.02
    20041225.1    5.20
    20050125.1    5.03
    20050225.1    5.02
    20050325.1    5.58
    20050425.1    4.98
    20050525.1    5.15
    20050625.1    4.85
    20050725.1    5.02
    20050825.1    4.82
    20050925.1    4.79
    20051025.1    4.94
    20051125.1    4.74
    20051225.1    4.89
    20060125.1    4.70
    20060225.1    4.67
    20060325.1    5.19
    20060425.1    4.61
    20060525.1    4.75
    20060625.1    4.70
    20060725.1    7.42
    20060825.1    7.20
    20060925.1    7.53
    20061025.1    7.80
    20061125.1    7.53
    20061225.1    7.87
    20070125.1    8.01
    20070225.1    7.78
    20070325.1    8.77
    20070425.1    7.88
    20070525.1    8.17
    20070625.1    8.11
    20070725.1    9.05
    20070825.1    8.80
    20070925.1    8.96
    20071025.1    9.27
    20071125.1    8.96
    20071225.1    9.40
    20080125.1    9.56
    20080225.1    9.56
    20080325.1    9.56
    20080425.1    9.56
    20080525.1    9.56
    20080625.1    9.56
    20080725.1    9.56
```

```
    20080825.1     9.56
    20080925.1     9.56
    20081025.1     9.56
    20081125.1     9.56
    20081225.1    0
!
 DEFINE DYNAMIC #CapBal12 = LOOKUP_TBL( "STEP", Curdate, "Cap12Notional", "CURDATE",
"Balance" )
!
 DEFINE DYNAMIC #CapBal12End = LOOKUP_TBL( "STEP", Curdate + 30, "Cap12Notional",
"CURDATE", "Balance" )
!
 DEFINE DYNAMIC #CapRate12 = LOOKUP_TBL( "STEP", Curdate, "Cap12RateSch", "CURDATE",
"Rate" )
!
!
 DEFINE TABLE "Cap22Notional" (36, 2) = "CURDATE" "Balance"
    20040825.1    0
    20040925.1    0
    20041025.1    0
    20041125.1    0
    20041225.1    0
    20050125.1    0
    20050225.1    0
    20050325.1    0
    20050425.1    0
    20050525.1    0
    20050625.1    0
    20050725.1    0
    20050825.1    0
    20050925.1    0
    20051025.1    0
    20051125.1    0
    20051225.1    0
    20060125.1    0
    20060225.1    0
    20060325.1    0
    20060425.1    0
    20060525.1    0
    20060625.1    0
    20060725.1    0
    20060825.1    0
    20060925.1    0
    20061025.1    0
    20061125.1    0
    20061225.1    0
    20070125.1    0
    20070225.1    0
    20070325.1    0
    20070425.1    0
    20070525.1    0
    20070625.1    0
    20070725.1    0
!
 DEFINE TABLE "Cap22RateSch" (36, 2) = "CURDATE" "Rate"
```

```
    20040825.1    0
    20040925.1    0
    20041025.1    0
    20041125.1    0
    20041225.1    0
    20050125.1    0
    20050225.1    0
    20050325.1    0
    20050425.1    0
    20050525.1    0
    20050625.1    0
    20050725.1    0
    20050825.1    0
    20050925.1    0
    20051025.1    0
    20051125.1    0
    20051225.1    0
    20060125.1    0
    20060225.1    0
    20060325.1    0
    20060425.1    0
    20060525.1    0
    20060625.1    0
    20060725.1    0
    20060825.1    0
    20060925.1    0
    20061025.1    0
    20061125.1    0
    20061225.1    0
    20070125.1    0
    20070225.1    0
    20070325.1    0
    20070425.1    0
    20070525.1    0
    20070625.1    0
    20070725.1    0
!
 DEFINE DYNAMIC #CapBal22 = LOOKUP_TBL( "STEP", Curdate, "Cap22Notional", "CURDATE",
"Balance" )
!
 DEFINE DYNAMIC #CapBal22End = LOOKUP_TBL( "STEP", Curdate + 30, "Cap22Notional",
"CURDATE", "Balance" )
!
 DEFINE DYNAMIC #CapRate22 = LOOKUP_TBL( "STEP", Curdate, "Cap22RateSch", "CURDATE",
"Rate" )
!
!
DEFINE TABLE "Cap32Notional" (32, 2) = "CURDATE" "Balance"
    20040825.1    0
    20040925.1    0
    20041025.1    0
    20041125.1    0
    20041225.1    0
    20050125.1    0
    20050225.1    0
```

```
    20050325.1    0
    20050425.1    0
    20050525.1    0
    20050625.1    0
    20050725.1    0
    20050825.1    0
    20050925.1    0
    20051025.1    0
    20051125.1    0
    20051225.1    0
    20060125.1    0
    20060225.1    0
    20060325.1    0
    20060425.1    0
    20060525.1    0
    20060625.1    0
    20060725.1    0
    20060825.1    0
    20060925.1    0
    20061025.1    0
    20061125.1    0
    20061225.1    0
    20070125.1    0
    20070225.1    0
    20070325.1    0
!
DEFINE TABLE "Cap32RateSch" (32, 2) = "CURDATE" "Rate"
    20040825.1    0
    20040925.1    0
    20041025.1    0
    20041125.1    0
    20041225.1    0
    20050125.1    0
    20050225.1    0
    20050325.1    0
    20050425.1    0
    20050525.1    0
    20050625.1    0
    20050725.1    0
    20050825.1    0
    20050925.1    0
    20051025.1    0
    20051125.1    0
    20051225.1    0
    20060125.1    0
    20060225.1    0
    20060325.1    0
    20060425.1    0
    20060525.1    0
    20060625.1    0
    20060725.1    0
    20060825.1    0
    20060925.1    0
    20061025.1    0
    20061125.1    0
```

```
    20061225.1    0
    20070125.1    0
    20070225.1    0
    20070325.1    0
!
 DEFINE DYNAMIC #CapBal32 = LOOKUP_TBL( "STEP", Curdate, "Cap32Notional", "CURDATE",
"Balance" )
!
 DEFINE DYNAMIC #CapBal32End = LOOKUP_TBL( "STEP", Curdate + 30, "Cap32Notional",
"CURDATE", "Balance" )
!
 DEFINE DYNAMIC #CapRate32 = LOOKUP_TBL( "STEP", Curdate, "Cap32RateSch", "CURDATE",
"Rate" )
!
!
 DEFINE #FloorCollat1       = 0.50% * #OrigCollBal1
 DEFINE #SpecOCTarg1        = 3.00% * #OrigCollBal1
 DEFINE STANDARDIZE OC_ACTUAL_VAL      GROUP 1        #OC1          = 761.12
 DEFINE STANDARDIZE OCT_INITVAL        GROUP 1 CONSTANT #InitOCTarg1    = 3.00% *
#OrigCollBal1
 DEFINE STANDARDIZE OCT_STEPDOWN_MONTH  GROUP 1 CONSTANT #StepDownDate1  = 37
 DEFINE STANDARDIZE OCT_STEPDOWN_FRAC   GROUP 1 CONSTANT #StepOCFrac1    = 0.06
 DEFINE STANDARDIZE EXCESS_INTEREST     GROUP 1        #XSSpread1      = 0
 DEFINE STANDARDIZE OCT_FLOOR          GROUP 1 CONSTANT #FloorOCTarg1   = #FloorCollat1
 DEFINE STANDARDIZE OCT_VAL            GROUP 1 DYNAMIC  #Octval1        = #SpecOCTarg1
!
 DEFINE #FloorCollat2       = 0.50% * #OrigCollBal2
 DEFINE #SpecOCTarg2        = 2.70% * #OrigCollBal2
 DEFINE STANDARDIZE OC_ACTUAL_VAL      GROUP 2        #OC2          = 238.88
 DEFINE STANDARDIZE OCT_INITVAL        GROUP 2 CONSTANT #InitOCTarg2    = 2.70% *
#OrigCollBal2
 DEFINE STANDARDIZE OCT_STEPDOWN_MONTH  GROUP 2 CONSTANT #StepDownDate2  = 37
 DEFINE STANDARDIZE OCT_STEPDOWN_FRAC   GROUP 2 CONSTANT #StepOCFrac2    = 0.054
 DEFINE STANDARDIZE EXCESS_INTEREST     GROUP 2        #XSSpread2      = 0
 DEFINE STANDARDIZE OCT_FLOOR          GROUP 2 CONSTANT #FloorOCTarg2   = #FloorCollat2
 DEFINE STANDARDIZE OCT_VAL            GROUP 2 DYNAMIC  #Octval2        = #SpecOCTarg2
!
 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 #NetRate2C_A  = ( COLL_I_MISC("COUPON","2C_A") ) /
COLL_PREV_BAL("2C_A") * 1200
 DEFINE DYNAMIC STICKY #NetRate2C_F  = ( COLL_I_MISC("COUPON","2C_F") ) /
COLL_PREV_BAL("2C_F") * 1200
 DEFINE DYNAMIC STICKY #NetRate2NC_A  = ( COLL_I_MISC("COUPON","2NC_A") ) /
COLL_PREV_BAL("2NC_A") * 1200
 DEFINE DYNAMIC STICKY #NetRate2NC_F  = ( COLL_I_MISC("COUPON","2NC_F") ) /
COLL_PREV_BAL("2NC_F") * 1200
!
!
```

```
 DEFINE DYNAMIC STICKY #NetRate2Actual360 = #Netrate2 * 30 / DAYS_DIFF(CURDATE ,
MONTHS_ADD(CURDATE,-1))
!
 DEFINE DYNAMIC STICKY #NetRate1Actual360 = #Netrate1 * 30 / DAYS_DIFF(CURDATE ,
MONTHS_ADD(CURDATE,-1))
!
 DEFINE TABLE "OC_CUMLOSS1" (37, 2) = "MONTH" "OC_CUMLOSS_FRAC1"
   37.1  0.0175
   38.1  0.01833333333
   39.1  0.01916666667
   40.1  0.02
   41.1  0.02083333333
   42.1  0.02166666667
   43.1  0.0225
   44.1  0.02333333333
   45.1  0.02416666667
   46.1  0.025
   47.1  0.02583333333
   48.1  0.02666666667
   49.1  0.0275
   50.1  0.028125
   51.1  0.02875
   52.1  0.029375
   53.1  0.03
   54.1  0.030625
   55.1  0.03125
   56.1  0.031875
   57.1  0.0325
   58.1  0.033125
   59.1  0.03375
   60.1  0.034375
   61.1  0.035
   62.1  0.03541667
   63.1  0.03583333
   64.1  0.03625
   65.1  0.03666667
   66.1  0.03708333
   67.1  0.0375
   68.1  0.03791667
   69.1  0.03833333
   70.1  0.03875
   71.1  0.03916667
   72.1  0.03958333
   400.1  0.04
!
 DEFINE TABLE "OC_CUMLOSS2" (37, 2) = "MONTH" "OC_CUMLOSS_FRAC2"
   37.1  0.0275
   38.1  0.02895833
   39.1  0.03041667
   40.1  0.031875
   41.1  0.03333333
   42.1  0.03479167
   43.1  0.03625
   44.1  0.03770833
   45.1  0.03916667
```

```
    46.1  0.040625
    47.1  0.04208333
    48.1  0.04354167
    49.1  0.045
    50.1  0.04604167
    51.1  0.04708333
    52.1  0.048125
    53.1  0.04916667
    54.1  0.05020833
    55.1  0.05125
    56.1  0.05229167
    57.1  0.05333333
    58.1  0.054375
    59.1  0.05541667
    60.1  0.05645833
    61.1  0.0575
    62.1  0.05779167
    63.1  0.05808333
    64.1  0.058375
    65.1  0.05866667
    66.1  0.05895833
    67.1  0.05925
    68.1  0.05954167
    69.1  0.05983333
    70.1  0.060125
    71.1  0.06041667
    72.1  0.06070833
    400.1  0.061
!
 DEFINE TABLE "NASAF" (5, 2) = "MONTH" "NAS_FRACAF"
    36.1  0%
    60.1  45%
    72.1  80%
    84.1  100%
    96.1  300%
!
 DEFINE DYNAMIC #AIOCA_SCHED = SCHED_AMOUNT("AIOCA")
 DEFINE DYNAMIC #AIOCA_ENDBAL = SCHED_AMOUNT("AIOCA_BAL")
!
 DEFINE DYNAMIC #AIOCF_SCHED = SCHED_AMOUNT("AIOCF")
 DEFINE DYNAMIC #AIOCF_ENDBAL = SCHED_AMOUNT("AIOCF_BAL")
!
 DEFINE DYNAMIC #AIONCA_SCHED = SCHED_AMOUNT("AIONCA")
 DEFINE DYNAMIC #AIONCA_ENDBAL = SCHED_AMOUNT("AIONCA_BAL")
!
 DEFINE DYNAMIC #AIONCF_SCHED = SCHED_AMOUNT("AIONCF")
 DEFINE DYNAMIC #AIONCF_ENDBAL = SCHED_AMOUNT("AIONCF_BAL")
!
!
!
TOLERANCE WRITEDOWN_0LOSS 9999999999999.00
TOLERANCE INTEREST       9999999999999.00
!
DEFINE DYNAMIC STICKY #MVBAL = BBAL("MV1#1", "MV2#1", "MV3#1", "MV4#1", "MV5#1",
"MV6#1")
```

```
DEFINE DYNAMIC STICKY #ALLBAL = BBAL("AV1", "AV2", "MV1", "MV2", "MV3", "MV4", "MV5","MV6")
DEFINE DYNAMIC STICKY #G1BAL =
BBAL("AF5#1","AF1#1","AF2#1","AF3#1","AF4#1","MF1#1","MF2#1","MF3#1","MF4#1","MF5#1","MF6#1
")

DEFINE DYNAMIC STICKY #CF_Percent  = (COLL_PREV_BAL("2C_A") + COLL_PREV_BAL("2C_F")) /
COLL_PREV_BAL(2)
DEFINE DYNAMIC STICKY #NCF_Percent = (COLL_PREV_BAL("2NC_A") +
COLL_PREV_BAL("2NC_F")) / COLL_PREV_BAL(2)

DEFINE DYNAMIC STICKY #NetWAC1   = COLL_I_MISC("COUPON",1) / #G1BAL * 1200
!!DEFINE DYNAMIC STICKY #NetWAC1   = COLL_I_MISC("COUPON",1) / COLL_PREV_BAL(1) * 1200
!!DEFINE DYNAMIC STICKY #NetWAC2   = MAX(0,(COLL_I_MISC("COUPON",2) -
OPTIMAL_INTPMT("AIOC#1","AIOC#2","AIONC#1","AIONC#2") -
OPTIMAL_INTPMT("AV1#1","AV2#1")) / BBAL("AV2" )) * 1200
!!DEFINE DYNAMIC STICKY #NetWAC2   = MAX(0,(COLL_I_MISC("COUPON",2) -
OPTIMAL_INTPMT("AIOC#1","AIOC#2","AIONC#1","AIONC#2") -
CAPPED_OPTIMAL_INTPMT("AV1#1","AV2#1")) / BBAL( "AV1", "AV2", "MV1", "MV2", "MV3", "MV4",
"MV5","MV6" )) * 1200
DEFINE DYNAMIC STICKY #NetWAC2   = IF CURMONTH LE 2 THEN 100 ELSE
MAX(0,(COLL_I_MISC("COUPON",2) - OPTIMAL_INTPMT("AIOC#1","AIOC#2","AIONC#1","AIONC#2"))
/ #ALLBAL) * 1200
DEFINE DYNAMIC STICKY #NetWAC2C  = ( COLL_I_MISC("COUPON","2C_A") +
COLL_I_MISC("COUPON","2C_F") - OPTIMAL_INTPMT("AIOC#1" , "AIOC#2") ) / (#ALLBAL ) * 1200
DEFINE DYNAMIC STICKY #NetWAC2NC = ( COLL_I_MISC("COUPON","2NC_A") +
COLL_I_MISC("COUPON","2NC_F") - OPTIMAL_INTPMT("AIONC#1","AIONC#2") ) / (BBAL("AV2") +
(#NCF_Percent * BBAL("AV2"))) * 1200

DEFINE DYNAMIC STICKY #AIO2_RATE   =
OPTIMAL_INTPMT("AIOC#1","AIOC#2","AIONC#1","AIONC#2") / COLL_PREV_BAL(2) * 1200
DEFINE DYNAMIC STICKY #AIO2C_RATE  = OPTIMAL_INTPMT("AIOC#1" , "AIOC#2") / (
BBAL("AV1")  ) * 1200
DEFINE DYNAMIC STICKY #AIO2NC_RATE = OPTIMAL_INTPMT("AIONC#1","AIONC#2") / (
COLL_PREV_BAL("2NC_A") + COLL_PREV_BAL("2NC_F") ) * 1200

DEFINE DYNAMIC STICKY #YM_AV1 =  ( OPTIMAL_INTPMT("CAP_IN12") / BBAL("AV1") * 1200 )
DEFINE DYNAMIC STICKY #YM_AV2 =  ( OPTIMAL_INTPMT("CAP_IN22") / BBAL("AV2") * 1200 )
DEFINE DYNAMIC STICKY #YM_MV = ( OPTIMAL_INTPMT("CAP_IN32") / #MVBAL * 1200 )
DEFINE DYNAMIC STICKY #YM_ALL =  ( OPTIMAL_INTPMT("CAP_IN12") / #ALLBAL * 1200 )

DEFINE DYNAMIC STICKY #AvlCapRate1   = IF CURMONTH LE 2 THEN 100 ELSE #NetWAC1
DEFINE DYNAMIC STICKY #AvlCapRate2C  = IF CURMONTH LE 2 THEN 100 ELSE #NetWAC2C +
#YM_AV1
DEFINE DYNAMIC STICKY #AvlCapRate2NC = IF CURMONTH LE 2 THEN 100 ELSE #NetWAC2NC +
#YM_AV2
DEFINE DYNAMIC STICKY #AvlCapRate2   = IF CURMONTH LE 2 THEN 100 ELSE #NetWAC2 +
#YM_MV

DEFINE DYNAMIC STICKY #AvlCapRate   = IF CURMONTH LE 2 THEN 100 ELSE #NetWAC2 +
#YM_ALL
DEFINE DYNAMIC STICKY #EffAvlCapRate   = IF CURMONTH LE 2 THEN 100 ELSE #AvlCapRate  *
30 / DAYS_DIFF(CURDATE , MONTHS_ADD(CURDATE,-1))

DEFINE DYNAMIC STICKY #EffAvlCapRate2   = IF CURMONTH LE 2 THEN 100 ELSE #AvlCapRate2
* 30 / DAYS_DIFF(CURDATE , MONTHS_ADD(CURDATE,-1))
```

```
DEFINE DYNAMIC STICKY #EffAvlCapRate2C  = IF CURMONTH LE 2 THEN 100 ELSE
#AvlCapRate2C  * 30 / DAYS_DIFF(CURDATE , MONTHS_ADD(CURDATE,-1))
DEFINE DYNAMIC STICKY #EffAvlCapRate2NC = IF CURMONTH LE 2 THEN 100 ELSE
#AvlCapRate2NC  * 30 / DAYS_DIFF(CURDATE , MONTHS_ADD(CURDATE,-1))


DEFINE DYNAMIC STICKY #NetWAC2_scale   = #NetWac2 * 30 / DAYS_DIFF(CURDATE ,
MONTHS_ADD(CURDATE,-1))

!!TEST Variables
DEFINE DYNAMIC STICKY #Int_Test                =              COLL_I_MISC("COUPON",2)
DEFINE DYNAMIC STICKY #IO_Test                 =
OPTIMAL_INTPMT("AIOC#1","AIOC#2","AIONC#1","AIONC#2")
DEFINE DYNAMIC STICKY #BBAL_test               =              BBAL("AV1", "AV2", "MV1", "MV2",
"MV3", "MV4", "MV5","MV6")



!!TRANCHE "#AvlCapRate1" SYMVAR
!!TRANCHE "#EffAvlCapRate2" SYMVAR
!!TRANCHE "#EffAvlCapRate2C" SYMVAR
!!TRANCHE "#EffAvlCapRate2NC" SYMVAR
!!TRANCHE "#TURBOPAID1" SYMVAR
!!TRANCHE "#TURBOPAID2" SYMVAR
!!TRANCHE "#NasF_CROSS_TEST" SYMVAR
!!TRANCHE "#NetWAC2C" SYMVAR
!!TRANCHE "#NetWAC2NC" SYMVAR
!!TRANCHE "#YM_AV1" SYMVAR
!!TRANCHE "#NetWac2" SYMVAR
!!TRANCHE "#EffAvlCapRate" SYMVAR
!!TRANCHE "#ALLBAL" SYMVAR
!!TRANCHE "#YM_ALL" SYMVAR
!!TRANCHE "#G1Bal" SYMVAR
!!TRANCHE "#NetWAC2_scale" SYMVAR
!
!!TRANCHE "#Int_Test" SYMVAR
!!TRANCHE "#IO_Test" SYMVAR
!!TRANCHE "#BBAL_test" SYMVAR
!
DEFINE DYNAMIC STICKY #CallTurbo = 1
!!Calculate : #Dummy14   =              #G2BAL
!!Calculate : #Dummy8    =              #AIO2_RATE
!!Calculate : #Dummy9    =              #YM_MV
 INITIAL INDEX   LIBOR_6MO       1.840
 INITIAL INDEX   CMT_1YR        1.990
 INITIAL INDEX   LIBOR_1MO       1.367
!
!
Tranche "CAP_IN12" PSEUDO HEDGE
  Block $ 0.01 at 1.367 FLOAT NOTIONAL WITH FORMULA BEGIN ( #CapBal12 ); END (
#CapBal12End ); _
       DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
       Delay 0  Dated 20040727  Next 20040825
   ((1 * MIN(9.56, LIBOR_1MO)) + (-1 * #CapRate12))
   0    999
```

```
!
Tranche "CAP_IN22" PSEUDO HEDGE
  Block $ 0.01 at 1.367 FLOAT NOTIONAL WITH FORMULA BEGIN ( #CapBal22 ); END (
#CapBal22End ); _
        DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
        Delay 0  Dated 20040727  Next 20040825
    ((1 * MIN(0.00, LIBOR_1MO)) + (-1 * #CapRate22))
    0    999
!
Tranche "CAP_IN32" PSEUDO HEDGE
  Block $ 0.01 at 1.367 FLOAT NOTIONAL WITH FORMULA BEGIN ( #CapBal32 ); END (
#CapBal32End ); _
        DAYCOUNT ACTUAL360 BUSINESS_DAY NONE FREQ M _
        Delay 0  Dated 20040727  Next 20040825
    ((1 * MIN(0.00, LIBOR_1MO)) + (-1 * #CapRate32))
    0    999
!
!
!
!
!
Tranche "AIOC" SEN_CPT_IO ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 58874000.00 at 4 GROUP "2C_A"  FREQ M NOTIONAL WITH FORMULA BEGIN ( MIN(
COLL_PREV_BAL("2C_A"), #AIOCA_SCHED )); _
                                END   ( MIN( COLL_BAL("2C_A"), #AIOCA_ENDBAL )); _
        DAYCOUNT 30360 BUSINESS_DAY NONE _
        Delay 24  Dated 20040701  Next 20040825
  Block 10376000.00 at 4 GROUP "2C_F"  FREQ M NOTIONAL WITH FORMULA BEGIN ( MIN(
COLL_PREV_BAL("2C_F"), #AIOCF_SCHED )); _
                                END   ( MIN( COLL_BAL("2C_F"), #AIOCF_ENDBAL )); _
        DAYCOUNT 30360 BUSINESS_DAY NONE _
        Delay 24  Dated 20040701  Next 20040825
!
Tranche "AIOCA" PSEUDO COMPONENT
  Block USE PCT   0.0 100.0 OF "AIOC#1"
!
Tranche "AIOCF" PSEUDO COMPONENT
  Block USE PCT   0.0 100.0 OF "AIOC#2"
!
Tranche "AIONC" SEN_CPT_IO ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 58903000.00 at 4 GROUP "2NC_A"  FREQ M NOTIONAL WITH FORMULA BEGIN ( MIN(
COLL_PREV_BAL("2NC_A"), #AIONCA_SCHED )); _
                                END   ( MIN( COLL_BAL("2NC_A"), #AIONCA_ENDBAL )); _
        DAYCOUNT 30360 BUSINESS_DAY NONE _
        Delay 24  Dated 20040701  Next 20040825
  Block 10347000.00 at 4 GROUP "2NC_F"  FREQ M NOTIONAL WITH FORMULA BEGIN ( MIN(
COLL_PREV_BAL("2NC_F"), #AIONCF_SCHED )); _
                                END   ( MIN( COLL_BAL("2NC_F"), #AIONCF_ENDBAL )); _
        DAYCOUNT 30360 BUSINESS_DAY NONE _
        Delay 24  Dated 20040701  Next 20040825
!
Tranche "AIONCA" PSEUDO COMPONENT
  Block USE PCT   0.0 100.0 OF "AIONC#1"
!
Tranche "AIONCF" PSEUDO COMPONENT
```

```
  Block USE PCT   0.0 100.0 OF "AIONC#2"
!
Tranche "AF5" MEZ_NAS_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 39753000.00 at 4.97 GROUP 1  FREQ M FLOAT _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT 30360 BUSINESS_DAY NONE _
       Delay 24  Dated 20040701  Next 20040825
   ( IF ((COLL_BAL("LAGMON_1",1) / #OrigCollBal1) < 20%) THEN 5.47 ELSE 4.97 )
   0   999
!
Tranche "AF1" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 181000000.00 at 1.567 GROUP 1  FREQ M FLOAT RESET M _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
       Delay 0  Dated 20040727  Next 20040825
    (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1",1) / #OrigCollBal1) < 20%) THEN 0.4 ELSE 0.2 ))
    0.20    999
!
Tranche "AF2" MEZ_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 107777000.00 at 4.06 GROUP 1  FREQ M FLOAT _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT 30360 BUSINESS_DAY NONE _
       Delay 24  Dated 20040701  Next 20040825
   ( 4.06 )
   0   999
!
Tranche "AF3" MEZ_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 19000000.00 at 5.12 GROUP 1  FREQ M FLOAT _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT 30360 BUSINESS_DAY NONE _
       Delay 24  Dated 20040701  Next 20040825
   ( 5.12 )
   0   999
!
Tranche "AF4" MEZ_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 50000000.00 at 5.5 GROUP 1  FREQ M FLOAT _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT 30360 BUSINESS_DAY NONE _
       Delay 24  Dated 20040701  Next 20040825
   ( IF ((COLL_BAL("LAGMON_1",1) / #OrigCollBal1) < 20%) THEN 6.0 else 5.5 )
   0   999
!
Tranche "MF1" SEN_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 22847000.00 at 5.5 GROUP 1  FREQ M FLOAT _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT 30360 BUSINESS_DAY NONE _
       Delay 24  Dated 20040701  Next 20040825
   ( IF ((COLL_BAL("LAGMON_1",1) / #OrigCollBal1) < 20%) THEN 6.0 else 5.5 )
   0   999
!
Tranche "MF2" MEZ_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 17135000.00 at 5.5 GROUP 1  FREQ M FLOAT _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT 30360 BUSINESS_DAY NONE _
       Delay 24  Dated 20040701  Next 20040825
```

```
  ( IF ((COLL_BAL("LAGMON_1",1) / #OrigCollBal1) < 20%) THEN 6.0 else 5.5 )
  0   999
!
Tranche "MF3" MEZ_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 5712000.00 at 5.5 GROUP 1  FREQ M FLOAT _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT 30360 BUSINESS_DAY NONE _
       Delay 24  Dated 20040701  Next 20040825
  ( IF ((COLL_BAL("LAGMON_1",1) / #OrigCollBal1) < 20%) THEN 6.0 else 5.5 )
  0   999
!
Tranche "MF4" MEZ_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 4569000.00 at 5.5 GROUP 1  FREQ M FLOAT _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT 30360 BUSINESS_DAY NONE _
       Delay 24  Dated 20040701  Next 20040825
  ( IF ((COLL_BAL("LAGMON_1",1) / #OrigCollBal1) < 20%) THEN 6.0 else 5.5 )
  0   999
!
Tranche "MF5" MEZ_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 4569000.00 at 5.5 GROUP 1  FREQ M FLOAT _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT 30360 BUSINESS_DAY NONE _
       Delay 24  Dated 20040701  Next 20040825
  ( IF ((COLL_BAL("LAGMON_1",1) / #OrigCollBal1) < 20%) THEN 6.0 else 5.5 )
  0   999
!
Tranche "MF6" MEZ_FIX_CAP ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 4568000.00 at 5.5 GROUP 1  FREQ M FLOAT _
       COUPONCAP 30360 NONE ( #AvlCapRate1 ); _
       DAYCOUNT 30360 BUSINESS_DAY NONE _
       Delay 24  Dated 20040701  Next 20040825
  ( IF ((COLL_BAL("LAGMON_1",1) / #OrigCollBal1) < 20%) THEN 6.0 else 5.5 )
  0   999
!
Tranche "AV1" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 309301000.00 at 1.647 GROUP "2C_A"  FREQ M FLOAT RESET M _
       COUPONCAP 30360 NONE ( #NetWAC2 ); _
       DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
       Delay 0  Dated 20040727  Next 20040825
   (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1",2) / #OrigCollBal2) < 20%) THEN 0.56 ELSE 0.28
))
   0.34    10
!
Tranche "AV2" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 309304000.00 at 1.667 GROUP "2C_F"  FREQ M FLOAT RESET M _
       COUPONCAP 30360 NONE ( #NetWAC2 ); _
       DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
       Delay 0  Dated 20040727  Next 20040825
   (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1",2) / #OrigCollBal2) < 20%) THEN 0.6 ELSE 0.3 ))
   0.30    10
!
Tranche "MV1" SEN_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 48300000.00 at 1.967 GROUP 2  FREQ M FLOAT RESET M _
       COUPONCAP 30360 NONE ( #NetWAC2 ); _
```

```
      DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
      Delay 0  Dated 20040727  Next 20040825
   (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1",2) / #OrigCollBal2) < 20%) THEN 0.9 ELSE 0.6 ))
   0    10
!
Tranche "MV2" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 39011000.00 at 2.567 GROUP 2  FREQ M FLOAT RESET M _
      COUPONCAP 30360 NONE ( #NetWAC2 ); _
      DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
      Delay 0  Dated 20040727  Next 20040825
   (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1",2) / #OrigCollBal2) < 20%) THEN 1.8 ELSE 1.2 ))
   0    10
!
Tranche "MV3" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 11146000.00 at 2.767 GROUP 2  FREQ M FLOAT RESET M _
      COUPONCAP 30360 NONE ( #NetWAC2 ); _
      DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
      Delay 0  Dated 20040727  Next 20040825
   (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1",2) / #OrigCollBal2) < 20%) THEN 2.1 ELSE 1.4 ))
   0    10
!
Tranche "MV4" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 11146000.00 at 3.217 GROUP 2  FREQ M FLOAT RESET M _
      COUPONCAP 30360 NONE ( #NetWAC2 ); _
      DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
      Delay 0  Dated 20040727  Next 20040825
   (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1",2) / #OrigCollBal2) < 20%) THEN 2.775 ELSE
1.85 ))
   0    10
!
Tranche "MV5" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 7431000.00 at 3.417 GROUP 2  FREQ M FLOAT RESET M _
      COUPONCAP 30360 NONE ( #NetWAC2 ); _
      DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
      Delay 0  Dated 20040727  Next 20040825
   (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1",2) / #OrigCollBal2) < 20%) THEN 3.075 ELSE
2.05 ))
   0    10
!
Tranche "MV6" MEZ_FLT ! PAID_DOWN_WHEN (COLL_BAL LT 0.01);
  Block 7430000.00 at 4.367 GROUP 2  FREQ M FLOAT RESET M _
      COUPONCAP 30360 NONE ( #NetWAC2 ); _
      DAYCOUNT ACTUAL360 BUSINESS_DAY NONE _
      Delay 0  Dated 20040727  Next 20040825
   (1 * LIBOR_1MO + ( IF ((COLL_BAL("LAGMON_1",2) / #OrigCollBal2) < 20%) THEN 4.5 ELSE 3 ))
   3.75    10
!
Tranche "R_1" SEN_RES
  Block 456930761.12 at 0 GROUP 1 NOTIONAL WITH GROUP 1 _
      DAYCOUNT 30360 BUSINESS_DAY NONE _
      FREQ M   Delay 24  Dated 20040701  Next 20040825
!
Tranche "R_PP_1" JUN_PEN_NO
  Block 456930761.12 at 0 GROUP 1 NOTIONAL WITH GROUP 1 _
      DAYCOUNT 30360 BUSINESS_DAY NONE _
```

```
      FREQ M   Delay 24  Dated 20040701  Next 20040825
!
Tranche "R_2" SEN_RES
  Block 743069238.88 at 0 GROUP 2 NOTIONAL WITH GROUP 2 _
      DAYCOUNT 30360 BUSINESS_DAY NONE _
      FREQ M   Delay 24  Dated 20040701  Next 20040825
!
Tranche "R_PP_2" JUN_PEN_NO
  Block 743069238.88 at 0 GROUP 2 NOTIONAL WITH GROUP 2 _
      DAYCOUNT 30360 BUSINESS_DAY NONE _
      FREQ M   Delay 24  Dated 20040701  Next 20040825
!
Tranche "MV" PSEUDO
  Block   USE PCT 100.0 100.0 of MV1#1
  Block   USE PCT 100.0 100.0 of MV2#1
  Block   USE PCT 100.0 100.0 of MV3#1
  Block   USE PCT 100.0 100.0 of MV4#1
  Block   USE PCT 100.0 100.0 of MV5#1
  Block   USE PCT 100.0 100.0 of MV6#1
!
Tranche "MF" PSEUDO
  Block   USE PCT 100.0 100.0 of MF1#1
  Block   USE PCT 100.0 100.0 of MF2#1
  Block   USE PCT 100.0 100.0 of MF3#1
  Block   USE PCT 100.0 100.0 of MF4#1
  Block   USE PCT 100.0 100.0 of MF5#1
  Block   USE PCT 100.0 100.0 of MF6#1
!
Tranche "AIO" PSEUDO
  Block   USE PCT 100.0 100.0 of AIOC#1
  Block   USE PCT 100.0 100.0 of AIOC#2
  Block   USE PCT 100.0 100.0 of AIONC#1
  Block   USE PCT 100.0 100.0 of AIONC#2
!
Tranche "F" PSEUDO
  Block   USE PCT 100.0 100.0 of MF1#1
  Block   USE PCT 100.0 100.0 of MF2#1
  Block   USE PCT 100.0 100.0 of MF3#1
  Block   USE PCT 100.0 100.0 of MF4#1
  Block   USE PCT 100.0 100.0 of MF5#1
  Block   USE PCT 100.0 100.0 of MF6#1
  Block   USE PCT 100.0 100.0 of AF5#1
  Block   USE PCT 100.0 100.0 of AF1#1
  Block   USE PCT 100.0 100.0 of AF2#1
  Block   USE PCT 100.0 100.0 of AF3#1
  Block   USE PCT 100.0 100.0 of AF4#1
!
Tranche "V" PSEUDO
  Block   USE PCT 100.0 100.0 of MV1#1
  Block   USE PCT 100.0 100.0 of MV2#1
  Block   USE PCT 100.0 100.0 of MV3#1
  Block   USE PCT 100.0 100.0 of MV4#1
  Block   USE PCT 100.0 100.0 of MV5#1
  Block   USE PCT 100.0 100.0 of MV6#1
  Block   USE PCT 100.0 100.0 of AV1#1
```

```
  Block    USE PCT 100.0 100.0 of AV2#1
!
Tranche "VIO" PSEUDO
  Block    USE PCT 100.0 100.0 of MV1#1
  Block    USE PCT 100.0 100.0 of MV2#1
  Block    USE PCT 100.0 100.0 of MV3#1
  Block    USE PCT 100.0 100.0 of MV4#1
  Block    USE PCT 100.0 100.0 of MV5#1
  Block    USE PCT 100.0 100.0 of MV6#1
  Block    USE PCT 100.0 100.0 of AV1#1
  Block    USE PCT 100.0 100.0 of AV2#1
  Block    USE PCT   0.0 100.0 of AIOC#1
  Block    USE PCT   0.0 100.0 of AIOC#2
  Block    USE PCT   0.0 100.0 of AIONC#1
  Block    USE PCT   0.0 100.0 of AIONC#2
!
 Tranche "#NetRate"            SYMVAR
!
 Tranche "#NetRate1"           SYMVAR
!
 Tranche "#NetRate1Actual360"       SYMVAR
!
 Tranche "#NetRate2"           SYMVAR
!
 Tranche "#NetRate2Actual360"       SYMVAR
!
 Tranche "#NetRate1F"          SYMVAR
!
 Tranche "#NetRate2C_A"          SYMVAR
!
 Tranche "#NetRate2C_F"          SYMVAR
!
 Tranche "#NetRate2NC_A"          SYMVAR
!
 Tranche "#NetRate2NC_F"          SYMVAR
!
 Tranche "#OC1"            SYMVAR
 Tranche "#SpecOCTarg1"    SYMVAR
!
 Tranche "#OC2"            SYMVAR
 Tranche "#SpecOCTarg2"    SYMVAR
!
Tranche "DEAL_PLUGIN" PSEUDO
  Block USE PCT 100.0 100.0 OF "AF5#1"
  Block USE PCT 100.0 100.0 OF "AF1#1"
  Block USE PCT 100.0 100.0 OF "AF2#1"
  Block USE PCT 100.0 100.0 OF "AF3#1"
  Block USE PCT 100.0 100.0 OF "AF4#1"
  Block USE PCT 100.0 100.0 OF "MF1#1"
  Block USE PCT 100.0 100.0 OF "MF2#1"
  Block USE PCT 100.0 100.0 OF "MF3#1"
  Block USE PCT 100.0 100.0 OF "MF4#1"
  Block USE PCT 100.0 100.0 OF "MF5#1"
  Block USE PCT 100.0 100.0 OF "MF6#1"
  Block USE PCT 100.0 100.0 OF "AV1#1"
```

```
  Block USE PCT 100.0 100.0 OF "AV2#1"
  Block USE PCT 100.0 100.0 OF "MV1#1"
  Block USE PCT 100.0 100.0 OF "MV2#1"
  Block USE PCT 100.0 100.0 OF "MV3#1"
  Block USE PCT 100.0 100.0 OF "MV4#1"
  Block USE PCT 100.0 100.0 OF "MV5#1"
  Block USE PCT 100.0 100.0 OF "MV6#1"
  Block USE PCT   0.0 100.0 OF "R_1#1"
  Block USE PCT   0.0 100.0 OF "R_2#1"
  Block USE PCT   0.0 100.0 OF "R_PP_1#1"
  Block USE PCT   0.0 100.0 OF "R_PP_2#1"
  Block USE PCT   0.0 100.0 OF "AIOC#1"
  Block USE PCT   0.0 100.0 OF "AIOC#2"
  Block USE PCT   0.0 100.0 OF "AIONC#1"
  Block USE PCT   0.0 100.0 OF "AIONC#2"
!
!
DEFINE PSEUDO_TRANCHE COLLAT _
  Delay 24 Dated 20040701 Next 20040825 Settle 20040727
DEFINE PSEUDO_TRANCHE COLLAT GROUP 1 _
  Delay 24 Dated 20040701 Next 20040825 Settle 20040727
DEFINE PSEUDO_TRANCHE COLLAT GROUP 2 _
  Delay 24 Dated 20040701 Next 20040825 Settle 20040727
!
 RESERVE_FUND "YmRsvFnd1"   FUNDING_FROM RULES
!
 RESERVE_FUND "YmRsvFnd2"   FUNDING_FROM RULES
!
 HEDGE "CAP12" _
            TYPE CAP _
            LEG  "FLT"   DEAL_RECEIVES   OPTIMAL_INTPMT  "CAP_IN12"
!
 HEDGE "CAP22" _
            TYPE CAP _
            LEG  "FLT"   DEAL_RECEIVES   OPTIMAL_INTPMT  "CAP_IN22"
!
 HEDGE "CAP32" _
            TYPE CAP _
            LEG  "FLT"   DEAL_RECEIVES   OPTIMAL_INTPMT  "CAP_IN32"
!

 CREDIT_SUPPORT_BASIS GROUP_DEAL
!
 CLASS "AF5"     NO_BUILD_TRANCHE _
            SHORTFALL_PAYBACK  COUPONCAP TRUE _
            SHORTFALL_EARN_INT COUPONCAP TRUE _
            = "AF5"
 CLASS "AF1"     NO_BUILD_TRANCHE _
            SHORTFALL_PAYBACK  COUPONCAP TRUE _
            SHORTFALL_EARN_INT COUPONCAP TRUE _
            = "AF1"
 CLASS "AF2"     NO_BUILD_TRANCHE _
            SHORTFALL_PAYBACK  COUPONCAP TRUE _
            SHORTFALL_EARN_INT COUPONCAP TRUE _
            = "AF2"
```

```
CLASS "AF3"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "AF3"
CLASS "AF4"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "AF4"
CLASS "MF1"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "MF1"
CLASS "MF2"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "MF2"
CLASS "MF3"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "MF3"
CLASS "MF4"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "MF4"
CLASS "MF5"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "MF5"
CLASS "MF6"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "MF6"
CLASS "RESID_1"  = "R_1#1" "R_PP_1#1"
CLASS "X"       = "AIOC#1" "AIONC#2" "AIONC#1" "AIOC#2"
CLASS "AV1"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "AV1"
CLASS "AV2"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "AV2"
CLASS "MV1"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "MV1"
CLASS "MV2"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "MV2"
CLASS "MV3"     NO_BUILD_TRANCHE _
          SHORTFALL_PAYBACK  COUPONCAP TRUE _
          SHORTFALL_EARN_INT COUPONCAP TRUE _
          = "MV3"
CLASS "MV4"     NO_BUILD_TRANCHE _
```

```
            SHORTFALL_PAYBACK  COUPONCAP TRUE _
            SHORTFALL_EARN_INT COUPONCAP TRUE _
            = "MV4"
 CLASS "MV5"      NO_BUILD_TRANCHE _
            SHORTFALL_PAYBACK  COUPONCAP TRUE _
            SHORTFALL_EARN_INT COUPONCAP TRUE _
            = "MV5"
 CLASS "MV6"      NO_BUILD_TRANCHE _
            SHORTFALL_PAYBACK  COUPONCAP TRUE _
            SHORTFALL_EARN_INT COUPONCAP TRUE _
            = "MV6"
 CLASS "RESID_2"   = "R_2#1" "R_PP_2#1"
 CLASS "AFNN" DISTRIB_CLASS PRORATA  WRITEDOWN_BAL PRORATA _
            = "AF1" "AF2" "AF3" "AF4"
 CLASS "AF"       WRITEDOWN_BAL PRORATA ALLOCATION _
            = "AF5" "AFNN"
 CLASS "AV"       DISTRIB_CLASS PRORATA  WRITEDOWN_BAL PRORATA ALLOCATION _
            = "AV1" "AV2"
!
!
 CLASS "GRP1" _
           WRITEDOWN_BAL RULES _
           DISTRIB_CLASS RULES _
           SHORTFALL_PAYBACK PRINCIPAL_LOSS TRUE _
           SHORTFALL_EARN_INT INTEREST TRUE _
            = "AF" "MF1" "MF2" "MF3" "MF4" "MF5" "MF6"  "RESID_1"
 CLASS "GRP2" _
           WRITEDOWN_BAL RULES _
           DISTRIB_CLASS RULES _
           SHORTFALL_PAYBACK PRINCIPAL_LOSS TRUE _
           SHORTFALL_EARN_INT INTEREST TRUE _
            = "X" "AV" "MV1" "MV2" "MV3" "MV4" "MV5" "MV6"  "RESID_2"
!
 CLASS "ROOT"  ROOT_LIST = "GRP1" "GRP2"
!
 GROUP 0      ROOT     = 1 2
!

!
 DEFINE PSEUDO_TRANCHE CLASS "AF"          Delay 24  Dated 20040701  Next 20040825
DAYCOUNT 30360 BUSINESS_DAY NONE
!
 DEFINE PSEUDO_TRANCHE CLASS "AV"          Delay 0  Dated 20040725  Next 20040825
DAYCOUNT ACTUAL360 BUSINESS_DAY NONE
!
 DEFINE PSEUDO_TRANCHE CLASS "AFNN"        Delay 24  Dated 20040701  Next 20040825
DAYCOUNT 30360 BUSINESS_DAY NONE
!
!
 CROSSOVER When 0
!
TRIGGER "StepUp-DlqRatio1" _
     FULL_NAME   "Group 1Step Up Delinquency Trigger" _
     ORIG_TESTVAL   0.000% _
     TESTVAL       ( #TrigDelinqFrac1); _
```

```
    ORIG_TARGETVAL  2% _
    TARGETVAL      (2%); _
    TRIGVAL        LODIFF
!
TRIGGER "StepUp-CumLoss1" _
    FULL_NAME   "Group 1Step Up Cumulative Loss Trigger" _
    ORIG_TESTVAL  0.000% _
    TESTVAL       ( #TrigCumLossFrac1); _
    ORIG_TARGETVAL  1.75% _
    TARGETVAL      (#CumLossShft1); _
    TRIGVAL        LODIFF
!
TRIGGER "StepUp-DlqEnh1" _
    FULL_NAME   "Group 1Step Up Enhancement Delinquency Trigger" _
    ORIG_TESTVAL  0.000% _
    TESTVAL       ( #TrigEnhFrac1); _
    ORIG_TARGETVAL  6.49997397575079% _
    TARGETVAL      (#ReqPerc1); _
    TRIGVAL        LODIFF
!
TRIGGER "STEPUP_TRIGGER1" _
    FULL_NAME   "Group 1Step Up Trigger" _
    DEFINITION "A Step Up Trigger exists, if_
;(1) the quotient of (A) the aggregate principal balance of all_
 mortgage loans 90 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 <=      %;_
              37        1.75%; _
              38        1.833333333%; _
              39        1.916666667%; _
              40        2%; _
              41        2.083333333%; _
              42        2.166666667%; _
              43        2.25%; _
              44        2.333333333%; _
              45        2.416666667%; _
              46        2.5%; _
              47        2.583333333%; _
              48        2.666666667%; _
              49        2.75%; _
              50        2.8125%; _
              51        2.875%; _
              52        2.9375%; _
              53        3%; _
              54        3.0625%; _
              55        3.125%; _
              56        3.1875%; _
              57        3.25%; _
              58        3.3125%; _
              59        3.375%; _
              60        3.4375%; _
              61        3.5%; _
              62        3.541667%; _
```

```
63        3.583333%; _
64        3.625%; _
65        3.666667%; _
66        3.708333%; _
67        3.75%; _
68        3.791667%; _
69        3.833333%; _
70        3.875%; _
71        3.916667%; _
72        3.958333%; _
400        4%; _
_
or;(3) the aggregate principal balance of all delinquent loans * 1_
as a percentage of the respective collateral balance exceeds :_
50% * 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 not occurred,_
or % of the balance when the trigger first occurred if a stepdown has occurred." _
    TRIGVAL FORMULA ( min(TRIGGER("StepUp-DlqRatio1","TRIGVAL"),
TRIGGER("StepUp-CumLoss1","TRIGVAL"), TRIGGER("StepUp-DlqEnh1","TRIGVAL")));
!
TRIGGER "StepUp-DlqRatio2" _
    FULL_NAME  "Group 2Step Up Delinquency Trigger" _
    ORIG_TESTVAL  0.000% _
    TESTVAL      ( #TrigDelinqFrac2); _
    ORIG_TARGETVAL  2% _
    TARGETVAL     (2%); _
    TRIGVAL       LODIFF
!
TRIGGER "StepUp-CumLoss2" _
    FULL_NAME  "Group 2Step Up Cumulative Loss Trigger" _
    ORIG_TESTVAL  0.000% _
    TESTVAL      ( #TrigCumLossFrac2); _
    ORIG_TARGETVAL  2.750000% _
    TARGETVAL     (#CumLossShft2); _
    TRIGVAL       LODIFF
!
TRIGGER "StepUp-DlqEnh2" _
    FULL_NAME  "Group 2Step Up Enhancement Delinquency Trigger" _
    ORIG_TESTVAL  0.000% _
    TESTVAL      ( #TrigEnhFrac2); _
    ORIG_TARGETVAL  6.95125789530113% _
    TARGETVAL     (#ReqPerc2); _
    TRIGVAL       LODIFF
!
TRIGGER "STEPUP_TRIGGER2" _
    FULL_NAME  "Group 2Step Up Trigger" _
    DEFINITION "A Step Up Trigger exists, if_
;(1) the quotient of (A) the aggregate principal balance of all_
mortgage loans 90 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 <=       %;_
              37        2.750000%; _
```

```
38        2.895833%; _
39        3.041667%; _
40        3.187500%; _
41        3.333333%; _
42        3.479167%; _
43        3.625000%; _
44        3.770833%; _
45        3.916667%; _
46        4.062500%; _
47        4.208333%; _
48        4.354167%; _
49        4.500000%; _
50        4.604167%; _
51        4.708333%; _
52        4.812500%; _
53        4.916667%; _
54        5.020833%; _
55        5.125000%; _
56        5.229167%; _
57        5.333333%; _
58        5.437500%; _
59        5.541667%; _
60        5.645833%; _
61        5.750000%; _
62        5.779167%; _
63        5.808333%; _
64        5.837500%; _
65        5.866667%; _
66        5.895833%; _
67        5.925000%; _
68        5.954167%; _
69        5.983333%; _
70        6.012500%; _
71        6.041667%; _
72        6.070833%; _
400       6.100000%; _
_
or;(3) the aggregate principal balance of all delinquent loans * 1_
as a percentage of the respective collateral balance exceeds :_
41.50% * 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 not occurred,_
or % of the balance when the trigger first occurred if a stepdown has occurred." _
    TRIGVAL FORMULA ( min(TRIGGER("StepUp-DlqRatio2","TRIGVAL"),
TRIGGER("StepUp-CumLoss2","TRIGVAL"), TRIGGER("StepUp-DlqEnh2","TRIGVAL")));
!
 OPTIONAL REDEMPTION:   "Cleanup1" _
            WHEN_EXPR ( COLL_BAL(1) / #OrigCollBal1 < 20% ); _
            TARGET GROUP "1F" _
            PRICE_P ( COLL_BAL(1) ); _
            DISTR_P RULES "OPTR_GROUP_1"
!
 OPTIONAL REDEMPTION:   "Cleanup2" _
            WHEN_EXPR ( COLL_BAL(2) / #OrigCollBal2 < 20% ); _
            TARGET GROUP "2C_A" & "2C_F" & "2NC_A" & "2NC_F" _
```

```
                PRICE_P ( COLL_BAL(2) ); _
                DISTR_P RULES "OPTR_GROUP_2"
!
!
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 #AF_Int =
{
-----------------------------------
     from :  CLASS ( "AF" )
      pay :  CLASS INTEREST PRO_RATA  ( "AF5"; "AFNN" )
-----------------------------------
-----------------------------------
     from :  CLASS ( "AFNN" )
      pay :  CLASS INTEREST PRO_RATA  ( "AF1"; "AF2"; "AF3"; "AF4" )
-----------------------------------
}
DEFINE MACRO BLOCK #AF_InS =
{
-----------------------------------
     from :  CLASS ( "AF" )
      pay :  CLASS INTSHORT PRO_RATA  ( "AF5"; "AFNN" )
-----------------------------------
-----------------------------------
     from :  CLASS ( "AFNN" )
      pay :  CLASS INTSHORT PRO_RATA  ( "AF1"; "AF2"; "AF3"; "AF4" )
-----------------------------------
}
DEFINE MACRO BLOCK #AF_Prn =
{
-----------------------------------
  calculate :  #NasCeilAF = MIN ( BBAL("AF5"), CASH_ACCOUNT("AF"), #NasFracAF * #NasShiftAF *
CASH_ACCOUNT("AF") )
-----------------------------------
 subject to :  CEILING ( #NasCeilAF )
     from :  CLASS ( "AF" )
      pay :  CLASS BALANCE SEQUENTIAL ("AF5")
-----------------------------------
     from :  CLASS ( "AF" )
      pay :  CLASS BALANCE SEQUENTIAL ("AFNN", "AF5")
-----------------------------------
!
     from :  CLASS ( "AFNN" )
      pay :  CLASS BALANCE SEQUENTIAL ( "AF1", "AF2", "AF3", "AF4" )
-----------------------------------
!
-----------------------------------
     from :  CLASS ( "AF5" )
      pay :  SEQUENTIAL ( "AF5#1" )
```

```
-----------------------------------
      from :  CLASS ( "AF1" )
       pay :  SEQUENTIAL ( "AF1#1" )
-----------------------------------
      from :  CLASS ( "AF2" )
       pay :  SEQUENTIAL ( "AF2#1" )
-----------------------------------
      from :  CLASS ( "AF3" )
       pay :  SEQUENTIAL ( "AF3#1" )
-----------------------------------
      from :  CLASS ( "AF4" )
       pay :  SEQUENTIAL ( "AF4#1" )
-----------------------------------
}
 DEFINE MACRO BLOCK #MF1_Prn =
{
-----------------------------------
      from :  CLASS ( "MF1" )
       pay :  SEQUENTIAL ( "MF1#1" )
-----------------------------------
}
 DEFINE MACRO BLOCK #MF2_Prn =
{
-----------------------------------
      from :  CLASS ( "MF2" )
       pay :  SEQUENTIAL ( "MF2#1" )
-----------------------------------
}
 DEFINE MACRO BLOCK #MF3_Prn =
{
-----------------------------------
      from :  CLASS ( "MF3" )
       pay :  SEQUENTIAL ( "MF3#1" )
-----------------------------------
}
 DEFINE MACRO BLOCK #MF4_Prn =
{
-----------------------------------
      from :  CLASS ( "MF4" )
       pay :  SEQUENTIAL ( "MF4#1" )
-----------------------------------
}
 DEFINE MACRO BLOCK #MF5_Prn =
{
-----------------------------------
      from :  CLASS ( "MF5" )
       pay :  SEQUENTIAL ( "MF5#1" )
-----------------------------------
}
 DEFINE MACRO BLOCK #MF6_Prn =
{
-----------------------------------
      from :  CLASS ( "MF6" )
       pay :  SEQUENTIAL ( "MF6#1" )
-----------------------------------
```

```
}
 DEFINE MACRO BLOCK #AV_Int =
{
-----------------------------------
      from :  CLASS ( "AV" )
       pay :  CLASS INTEREST PRO_RATA  ( "AV1"; "AV2" )
-----------------------------------
}
 DEFINE MACRO BLOCK #AV_InS =
{
-----------------------------------
      from :  CLASS ( "AV" )
       pay :  CLASS INTSHORT PRO_RATA  ( "AV1"; "AV2" )
-----------------------------------
}
 DEFINE MACRO BLOCK #AV_Prn[1] =
{
-----------------------------------
      when :  IS_TRUE( {#1} )
  calculate :  #SeniorPrinc2        = #PrincPmt2 / #DistribAmt2 * #ClassAVPDA
  calculate :  #SeniorXtraP2        = #ClassAVPDA - #SeniorPrinc2
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!  Update for Conf/Non-Conf   !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  calculate :  #SeniorPDA2C         = MIN( BBAL("AV1"), (#SeniorPrinc2 * #PrincFrac2C) +
(#SeniorXtraP2 * #XtraPFrac2C))
  calculate :  #SeniorPDA2NC        = MIN( BBAL("AV2"), (#SeniorPrinc2 * #PrincFrac2NC) +
(#SeniorXtraP2 * #XtraPFrac2NC))
-----------------------------------
      from :  SUBACCOUNT ( #SeniorPDA2C, CLASS "AV" )
       pay :  CLASS BALANCE SEQUENTIAL ( "AV1" )
-----------------------------------
      from :  SUBACCOUNT ( #SeniorPDA2NC, CLASS "AV" )
       pay :  CLASS BALANCE SEQUENTIAL ( "AV2" )
-----------------------------------
      from :  CLASS ( "AV" )
       pay :  CLASS BALANCE PRO_RATA ( "AV1"; "AV2" )
-----------------------------------
!
-----------------------------------
      from :  CLASS ( "AV1" )
       pay :  SEQUENTIAL ( "AV1#1" )
-----------------------------------
      from :  CLASS ( "AV2" )
       pay :  SEQUENTIAL ( "AV2#1" )
-----------------------------------
}
 DEFINE MACRO BLOCK #MV1_Prn =
{
-----------------------------------
      from :  CLASS ( "MV1" )
       pay :  SEQUENTIAL ( "MV1#1" )
-----------------------------------
}
```

```
DEFINE MACRO BLOCK #MV2_Prn =
{
-----------------------------------
      from :  CLASS ( "MV2" )
       pay :  SEQUENTIAL ( "MV2#1" )
-----------------------------------
}
DEFINE MACRO BLOCK #MV3_Prn =
{
-----------------------------------
      from :  CLASS ( "MV3" )
       pay :  SEQUENTIAL ( "MV3#1" )
-----------------------------------
}
DEFINE MACRO BLOCK #MV4_Prn =
{
-----------------------------------
      from :  CLASS ( "MV4" )
       pay :  SEQUENTIAL ( "MV4#1" )
-----------------------------------
}
DEFINE MACRO BLOCK #MV5_Prn =
{
-----------------------------------
      from :  CLASS ( "MV5" )
       pay :  SEQUENTIAL ( "MV5#1" )
-----------------------------------
}
DEFINE MACRO BLOCK #MV6_Prn =
{
-----------------------------------
      from :  CLASS ( "MV6" )
       pay :  SEQUENTIAL ( "MV6#1" )
-----------------------------------
}
!
!
CMO Block Payment Rules
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! APPLY 100% of PREPAYPENALTY TO YMRSVFND!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
-----------------------------------
      from :  CASH_ACCOUNT (100, GROUP 1)
 subject to :  CEILING ( (COLL("PREPAYPENALTY",1) * (100%)) )
       pay :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
-----------------------------------
      from :  CASH_ACCOUNT (100, GROUP 2)
 subject to :  CEILING ( (COLL("PREPAYPENALTY",2) * (100%)) )
       pay :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
-----------------------------------
-----------------------------------
  calculate : #HedgePaySave2 = 0.00
-----------------------------------
  calculate : #Coll_I1        = COLL_I(1) - COLL_YM(1) + CF_REINV_INCOME * COLL_CF(1)/COLL_CF
```

```
  calculate : #Coll_P1        = COLL_P(1)
-----------------------------------
  calculate : #Coll_I2        = COLL_I(2) - COLL_YM(2) + CF_REINV_INCOME * COLL_CF(2)/COLL_CF
+ HEDGE("CAP12","ACTUAL_PMT") + HEDGE("CAP22","ACTUAL_PMT") +
HEDGE("CAP32","ACTUAL_PMT")
  calculate : #Coll_P2        = COLL_P(2)
-----------------------------------
  calculate :  #Princ1          = COLL_P(1)
!
  calculate :  #Interest1        = COLL_I(1)
!
  calculate :  #Princ2          = COLL_P(2)
!
  calculate :  #Interest2        = COLL_I(2)
!
  calculate :  #PrevSpecOC1        = #SpecOCTarg1
!
  calculate :  #CurrentOC1        = MAX( 0, COLL_BAL(1) - (BBAL("AF5#1", "AF1#1", "AF2#1", "AF3#1",
"AF4#1", "MF1#1", "MF2#1", "MF3#1", "MF4#1", "MF5#1", "MF6#1") - #Princ1))
!
  calculate :  #XSSpread1        = MAX( 0, #Interest1 - ( COLL_YM(1) * (100% - 60%)) -
OPTIMAL_INTPMT("GRP1") - INTSHORT_ACCUM("GRP1") + COUPONCAP_SHORTFALL("GRP1") )
!
  calculate :  #FloorOCTotal1      = #FloorOCTarg1
!
  calculate :  #StepOCTarg1        = COLL_BAL(1) * #StepOCFrac1
!
  calculate :  #StepDownDatePass1   = CURMONTH GE #StepDownDate1
!
!!!********** BEGINNING OF SENIOR ENHANCEMENT PCT CALCULATION **********
!!! ASSUME STEPDOWN IN ORDER TO CALCULATE SENIOR ENHANCMENT PCT
  calculate :  #SpecOCTarg1        = MAX( MIN( #InitOCTarg1, #StepOCTarg1 ) , #FloorOCTotal1 )
!
  calculate :  #SpecOCTarg1        = MIN( #SpecOCTarg1, COLL_BAL(1) )
!
  calculate :  #SpecOCTarg1        = #Octval1
!
  calculate :  #OCDeficiency1      = MAX(0, #SpecOCTarg1 - #CurrentOC1)
!
  calculate :  #OCSurplus1        = MINMAX(0, #CurrentOC1 - #SpecOCTarg1, COLL_P(1))
!
  calculate :  #PrincPmt1        = MAX(0, COLL_P(1) - #OCSurplus1)
!
  calculate :  #XSIntRem1        = MAX( 0, #Interest1 - ( COLL_YM(1) * (100% - 60%)) -
OPTIMAL_INTPMT("GRP1") - INTSHORT_ACCUM("GRP1") + #OCSurplus1 +
COUPONCAP_SHORTFALL("GRP1"))
  calculate :  #XS1 = #Interest1
  calculate :  #XS2 = - ( COLL_YM(1) * (100% - 60%))
  calculate :  #XS3 = - OPTIMAL_INTPMT("GRP1")
  calculate :  #XS4 = - INTSHORT_ACCUM("GRP1")
  calculate :  #XS5 = #OCSurplus1
  calculate :  #XS6 = COUPONCAP_SHORTFALL("GRP1")
!
  calculate :  #SubDefic1        = MAX ( 0, ( BBAL("GRP1") - #Princ1 ) - COLL_BAL(1) )
!
```

```
  calculate :  #AddPrinc1          = MIN( #XSIntRem1, #SubDefic1 )
  calculate :  #XSIntRem1          = MAX( 0, #XSIntRem1 - #AddPrinc1 )
!
  calculate :  #XtraPDA1           = MIN( #OCDeficiency1, #XSIntRem1 )
  calculate :  #XSIntRem1          = MAX( 0, #XSIntRem1 - #XtraPDA1 )
!
!
  calculate :  #DistribAmt1        = #PrincPmt1 + #AddPrinc1 + #XtraPDA1
!
  calculate :  #ClassAFPDA         = BBAL("AF5", "AF1", "AF2", "AF3", "AF4") _
                        - MIN(COLL_BAL(1) - #FloorOCTotal1, #AFTargPct * COLL_BAL(1))
  calculate :  #ClassAFPDA         = MAX( 0.0, MIN(BBAL("AF5", "AF1", "AF2", "AF3", "AF4"),
#ClassAFPDA ))
  calculate :  #ClassAFPDA         = MAX( 0, MIN( #ClassAFPDA, #DistribAmt1 ) )

!
!
!!!********** END OF SENIOR ENHANCEMENT PCT CALCULATION **********
!
  calculate :  #SenEnhancePct1     = (COLL_BAL(1) - (BBAL("AF") - #ClassAFPDA )) / COLL_BAL(1)
!
  calculate :  #StepDownBal1       = (#SenEnhancePct1 - #SpecSenEnhPct1) + 1E-8 GE 0.00
!
  calculate :  #StepDown1          = #StepDown1 OR ( BBAL("AF") LT 0.01 ) OR ( #StepDownDatePass1
AND #StepDownBal1 )
!
  calculate :  #TrigDelinqFrac1    = AVG_COLL("RATE",-1,3,1,1)
!
  calculate :  #ReqPerc1           = 0.43 * (COLL_PREV_BAL(1) - BBAL("AF") ) / COLL_PREV_BAL(1)
!
  calculate :  #TrigEnhFrac1       = 1 * AVG_COLL("RATE",-1,2,1,1)
!
  calculate :  #CumLossShft1       = LOOKUP_TBL( "STEP",  CURMONTH    , "OC_CUMLOSS1",
"MONTH", "OC_CUMLOSS_FRAC1" )
  calculate :  #TrigCumLossFrac1   = DELINQ_LOSS_ACCUM(1) / #OrigCollBal1
!
  calculate :  #TrigEvent1         = TRIGGER("STEPUP_TRIGGER1")
!
  calculate :  #TrigOCTargPre1     = #PrevSpecOC1
!
  calculate :  #TrigOCTargPost1    = #PrevSpecOC1
!
  calculate :  #SpecOCTarg1        = IF #StepDown1 _
                        THEN IF #TrigEvent1 _
                           THEN MAX( MIN( #InitOCTarg1, #StepOCTarg1 ) , #TrigOCTargPost1,
#FloorOCTotal1 ) _
                           ELSE MAX( MIN( #InitOCTarg1, #StepOCTarg1 ) , #FloorOCTotal1 ) _
                        ELSE IF #TrigEvent1 _
                           THEN MAX( #InitOCTarg1, #TrigOCTargPre1, #FloorOCTotal1 ) _
                           ELSE MAX( #InitOCTarg1, #FloorOCTotal1 )
!
  calculate :  #SpecOCTarg1        = MIN( #SpecOCTarg1, COLL_BAL(1) )
!
  calculate :  #SpecOCTarg1        = #Octval1
!
```

```
  calculate :  #OCDeficiency1      = MAX(0, #SpecOCTarg1 - #CurrentOC1)
!
  calculate :  #OCSurplus1         = MINMAX(0, #CurrentOC1 - #SpecOCTarg1, COLL_P(1))
!
  calculate :  #PrincPmt1          = MAX(0, COLL_P(1) - #OCSurplus1)
!
  calculate :  #PrevSpecOC2        = #SpecOCTarg2
!
  calculate :  #CurrentOC2         = MAX( 0, COLL_BAL(2) - (BBAL("AV1#1", "AV2#1", "MV1#1",
"MV2#1", "MV3#1", "MV4#1", "MV5#1", "MV6#1") - #Princ2))
!
  calculate :  #XSSpread2          = MAX( 0, #Interest2 - ( COLL_YM(2) * (100% - 60%)) -
OPTIMAL_INTPMT("GRP2") - INTSHORT_ACCUM("GRP2") + COUPONCAP_SHORTFALL("GRP2") )
!
  calculate :  #FloorOCTotal2      = #FloorOCTarg2
!
  calculate :  #StepOCTarg2        = COLL_BAL(2) * #StepOCFrac2
!
  calculate :  #StepDownDatePass2  = CURMONTH GE #StepDownDate2
!
!!!********** BEGINNING OF SENIOR ENHANCEMENT PCT CALCULATION **********
!!! ASSUME STEPDOWN IN ORDER TO CALCULATE SENIOR ENHANCMENT PCT
  calculate :  #SpecOCTarg2        = MAX( MIN( #InitOCTarg2, #StepOCTarg2 ) , #FloorOCTotal2 )
!
  calculate :  #SpecOCTarg2        = MIN( #SpecOCTarg2, COLL_BAL(2) )
!
  calculate :  #SpecOCTarg2        = #Octval2
!
  calculate :  #OCDeficiency2      = MAX(0, #SpecOCTarg2 - #CurrentOC2)
!
  calculate :  #OCSurplus2         = MINMAX(0, #CurrentOC2 - #SpecOCTarg2, COLL_P(2))
!
  calculate :  #PrincPmt2          = MAX(0, COLL_P(2) - #OCSurplus2)
!
  calculate :  #XSIntRem2          = MAX( 0, #Interest2 - ( COLL_YM(2) * (100% - 60%)) -
OPTIMAL_INTPMT("GRP2") - INTSHORT_ACCUM("GRP2") + #OCSurplus2 +
COUPONCAP_SHORTFALL("GRP2"))
!
  calculate :  #SubDefic2          = MAX ( 0, ( BBAL("GRP2") - #Princ2 ) - COLL_BAL(2) )
!
  calculate :  #AddPrinc2          = MIN( #XSIntRem2, #SubDefic2 )
  calculate :  #XSIntRem2          = MAX( 0, #XSIntRem2 - #AddPrinc2 )
!
  calculate :  #XtraPDA2           = MIN( #OCDeficiency2, #XSIntRem2 )
  calculate :  #XSIntRem2          = MAX( 0, #XSIntRem2 - #XtraPDA2 )
!
!
  calculate :  #DistribAmt2        = #PrincPmt2 + #AddPrinc2 + #XtraPDA2
!
  calculate :  #ClassAVPDA         = BBAL("AV1", "AV2" ) _
                        - MIN(COLL_BAL(2) - #FloorOCTotal2, #AVTargPct * COLL_BAL(2))
  calculate :  #ClassAVPDA         = MAX( 0.0, MIN(BBAL("AV1", "AV2") , #ClassAVPDA ))
  calculate :  #ClassAVPDA         = MAX( 0, MIN( #ClassAVPDA, #DistribAmt2 ) )
!
!
```

```
!!!********** END OF SENIOR ENHANCEMENT PCT CALCULATION **********
!
  calculate : #SenEnhancePct2      = (COLL_BAL(2) - (BBAL("AV") - #ClassAVPDA )) / COLL_BAL(2)
!
  calculate : #StepDownBal2        = (#SenEnhancePct2 - #SpecSenEnhPct2) + 1E-8 GE 0.00
!
  calculate : #StepDown2           = #StepDown2 OR ( BBAL("AV") LT 0.01 ) OR ( #StepDownDatePass2
AND #StepDownBal2 )
!
  calculate : #TrigDelinqFrac2     = AVG_COLL("RATE",-1,3,1,2)
!
  calculate : #ReqPerc2            = 0.43 * (COLL_PREV_BAL(2) - BBAL("AV") ) / COLL_PREV_BAL(2)
!
  calculate : #TrigEnhFrac2        = 1 * AVG_COLL("RATE",-1,2,1,2)
!
  calculate : #CumLossShft2        = LOOKUP_TBL( "STEP",  CURMONTH     , "OC_CUMLOSS2",
"MONTH", "OC_CUMLOSS_FRAC2" )
  calculate : #TrigCumLossFrac2    = DELINQ_LOSS_ACCUM(2) / #OrigCollBal2
!
  calculate : #TrigEvent2          = TRIGGER("STEPUP_TRIGGER2")
!
  calculate : #TrigOCTargPre2      = #PrevSpecOC2
!
  calculate : #TrigOCTargPost2     = #PrevSpecOC2
!
  calculate : #SpecOCTarg2         = IF #StepDown2 _
                        THEN IF #TrigEvent2 _
                           THEN MAX( MIN( #InitOCTarg2, #StepOCTarg2 ) , #TrigOCTargPost2,
#FloorOCTotal2 ) _
                           ELSE MAX( MIN( #InitOCTarg2, #StepOCTarg2 ) , #FloorOCTotal2 ) _
                        ELSE IF #TrigEvent2 _
                           THEN MAX( #InitOCTarg2, #TrigOCTargPre2, #FloorOCTotal2 ) _
                           ELSE MAX( #InitOCTarg2, #FloorOCTotal2 )
!
  calculate : #SpecOCTarg2         = MIN( #SpecOCTarg2, COLL_BAL(2) )
!
  calculate : #SpecOCTarg2         = #Octval2
!
  calculate : #OCDeficiency2       = MAX(0, #SpecOCTarg2 - #CurrentOC2)
!
  calculate : #OCSurplus2          = MINMAX(0, #CurrentOC2 - #SpecOCTarg2, COLL_P(2))
!
  calculate : #PrincPmt2           = MAX(0, COLL_P(2) - #OCSurplus2)
!
!
  calculate : #NasShiftAF          = LOOKUP_TBL( "STEP", CURMONTH , "NASAF", "MONTH",
"NAS_FRACAF" )
!
  calculate : #NasFracAF           = BBAL("AF5")/ BBAL("AF5", "AFNN")
!
  calculate : #XSIntRem1           = MAX( 0, #Interest1 - ( COLL_YM(1) * (100% - 60%)) -
OPTIMAL_INTPMT("GRP1") - INTSHORT_ACCUM("GRP1") + #OCSurplus1 +
COUPONCAP_SHORTFALL("GRP1"))
!
  calculate : #XSIntRem2           = MAX( 0, #Interest2 - ( COLL_YM(2) * (100% - 60%)) -
```

```
OPTIMAL_INTPMT("GRP2") - INTSHORT_ACCUM("GRP2") + #OCSurplus2 +
COUPONCAP_SHORTFALL("GRP2"))
!
  calculate : #SubDefic1        = MAX ( 0, ( BBAL("GRP1") - #Princ1 ) - COLL_BAL(1) )
!
  calculate : #AddPrinc1        = MIN( #XSIntRem1, #SubDefic1 )
  calculate : #XSIntRem1         = MAX( 0, #XSIntRem1 - #AddPrinc1 )
!
  calculate : #SubDefic2        = MAX ( 0, ( BBAL("GRP2") - #Princ2 ) - COLL_BAL(2) )
!
  calculate : #AddPrinc2        = MIN( #XSIntRem2, #SubDefic2 )
  calculate : #XSIntRem2         = MAX( 0, #XSIntRem2 - #AddPrinc2 )
!
  calculate : #XtraPDA1         = MIN( #OCDeficiency1, #XSIntRem1 )
  calculate : #XSIntRem1         = MAX( 0, #XSIntRem1 - #XtraPDA1 )
!
  calculate : #XtraPDA2         = MIN( #OCDeficiency2, #XSIntRem2 )
  calculate : #XSIntRem2         = MAX( 0, #XSIntRem2 - #XtraPDA2 )
!
!!LIMITS BASIS RISK UNTIL AFTER CROSS
  calculate : #XSIntRem1                                    = MAX( 0, #XSIntRem1 -
COUPONCAP_ACCUM_SHORTFALL("GRP1") )
  calculate : #XSIntRem2                                    = MAX( 0, #XSIntRem2 -
COUPONCAP_ACCUM_SHORTFALL("GRP2") )
!
  calculate : #MoreDefic1       = MAX( 0, #SubDefic1 - #AddPrinc1 )
!
  calculate : #MoreDefic2       = MAX( 0, #SubDefic2 - #AddPrinc2 )
!
  calculate : #AddPrinc1C        = MIN(( #XSIntRem2 ) * ( #MoreDefic1 / ( #MoreDefic1 + #MoreDefic2)),
#MoreDefic1 )
  calculate : #AddPrinc2C        = MIN(( #XSIntRem1 ) * ( #MoreDefic2 / ( #MoreDefic1 + #MoreDefic2)),
#MoreDefic2 )
!
  calculate : #XSIntOut1        = MIN(( #AddPrinc2C ) * ( #XSIntRem1 / ( #XSIntRem1 + #XSIntRem2)),
#XSIntRem1 )
  calculate : #XSIntOut2        = MIN(( #AddPrinc1C ) * ( #XSIntRem2 / ( #XSIntRem1 + #XSIntRem2)),
#XSIntRem2 )
!
  calculate : #XSIntRem1         = MAX( 0, #XSIntRem1 - #XSIntOut1 )
  calculate : #XSIntRem2         = MAX( 0, #XSIntRem2 - #XSIntOut2 )
!
  calculate : #NeedPDA1C          = MAX( 0, #OCDeficiency1- #XtraPDA1 )
  calculate : #NeedPDA2C          = MAX( 0, #OCDeficiency2- #XtraPDA2 )
!
  calculate : #XtraPDA1C         = MIN( #NeedPDA1C , (#XSIntRem2 ) * ( #NeedPDA1C / (
#NeedPDA1C + #NeedPDA2C)) )
  calculate : #XtraPDA2C         = MIN( #NeedPDA2C , (#XSIntRem1 ) * ( #NeedPDA2C / (
#NeedPDA2C + #NeedPDA1C)) )
!
  calculate : #XSIntOut1        = MIN(( #XtraPDA2C ) * ( #XSIntRem1 / ( #XSIntRem1 + #XSIntRem2)),
#XSIntRem1 )
  calculate : #XSIntOut2        = MIN(( #XtraPDA1C ) * ( #XSIntRem2 / ( #XSIntRem1 + #XSIntRem2)),
#XSIntRem2 )
!
```

```
  calculate :  #XSIntRem1          = MAX( 0, #XSIntRem1 - #XSIntOut1 )
  calculate :  #XSIntRem2          = MAX( 0, #XSIntRem2 - #XSIntOut2 )
!
!
  calculate :  #DistribAmt1        = #PrincPmt1 + #AddPrinc1 + #XtraPDA1 + #AddPrinc1C + #XtraPDA1C
  calculate :  #DistribAmt2        = #PrincPmt2 + #AddPrinc2 + #XtraPDA2 + #AddPrinc2C + #XtraPDA2C
!
  calculate :  #ClassAFPDA         = IF (#TrigEvent1 OR (#StepDown1 EQ 0.0)) _
                          THEN #DistribAmt1 _
                          ELSE BBAL("AF5", "AF1", "AF2", "AF3", "AF4") _
                          - MIN(COLL_BAL(1) - #FloorOCTotal1, #AFTargPct * COLL_BAL(1))
  calculate :  #ClassAFPDA         = MAX( 0.0, MIN(BBAL("AF5", "AF1", "AF2", "AF3", "AF4"),
#ClassAFPDA ))
  calculate :  #ClassAFPDA         = MAX( 0, MIN( #ClassAFPDA, #DistribAmt1 ) )
!
!
  calculate :  #ClassMF1PDA        = IF (#TrigEvent1 OR (#StepDown1 EQ 0.0)) _
                          THEN #DistribAmt1 - #ClassAFPDA _
                          ELSE BBAL("AF5", "AF1", "AF2", "AF3", "AF4", "MF1") - #ClassAFPDA _
                          - MIN(COLL_BAL(1) - #FloorOCTotal1, #MF1TargPct * COLL_BAL(1))
  calculate :  #ClassMF1PDA        = MAX( 0.0, MIN(BBAL("MF1"), #ClassMF1PDA ))
  calculate :  #ClassMF1PDA        = MAX( 0, MIN( #ClassMF1PDA, #DistribAmt1 - #ClassAFPDA ) )
!
!
  calculate :  #ClassMF2PDA        = IF (#TrigEvent1 OR (#StepDown1 EQ 0.0)) _
                          THEN #DistribAmt1 - #ClassAFPDA - #ClassMF1PDA _
                          ELSE BBAL("AF5", "AF1", "AF2", "AF3", "AF4", "MF1", "MF2") - #ClassAFPDA -
#ClassMF1PDA _
                          - MIN(COLL_BAL(1) - #FloorOCTotal1, #MF2TargPct * COLL_BAL(1))
  calculate :  #ClassMF2PDA        = MAX( 0.0, MIN(BBAL("MF2"), #ClassMF2PDA ))
  calculate :  #ClassMF2PDA        = MAX( 0, MIN( #ClassMF2PDA, #DistribAmt1 - #ClassAFPDA -
#ClassMF1PDA ) )
!
!
  calculate :  #ClassMF3PDA        = IF (#TrigEvent1 OR (#StepDown1 EQ 0.0)) _
                          THEN #DistribAmt1 - #ClassAFPDA - #ClassMF1PDA - #ClassMF2PDA _
                          ELSE BBAL("AF5", "AF1", "AF2", "AF3", "AF4", "MF1", "MF2", "MF3") -
#ClassAFPDA - #ClassMF1PDA - #ClassMF2PDA _
                          - MIN(COLL_BAL(1) - #FloorOCTotal1, #MF3TargPct * COLL_BAL(1))
  calculate :  #ClassMF3PDA        = MAX( 0.0, MIN(BBAL("MF3"), #ClassMF3PDA ))
  calculate :  #ClassMF3PDA        = MAX( 0, MIN( #ClassMF3PDA, #DistribAmt1 - #ClassAFPDA -
#ClassMF1PDA - #ClassMF2PDA ) )
!
!
  calculate :  #ClassMF4PDA        = IF (#TrigEvent1 OR (#StepDown1 EQ 0.0)) _
                          THEN #DistribAmt1 - #ClassAFPDA - #ClassMF1PDA - #ClassMF2PDA -
#ClassMF3PDA _
                          ELSE BBAL("AF5", "AF1", "AF2", "AF3", "AF4", "MF1", "MF2", "MF3", "MF4") -
#ClassAFPDA - #ClassMF1PDA - #ClassMF2PDA - #ClassMF3PDA _
                          - MIN(COLL_BAL(1) - #FloorOCTotal1, #MF4TargPct * COLL_BAL(1))
  calculate :  #ClassMF4PDA        = MAX( 0.0, MIN(BBAL("MF4"), #ClassMF4PDA ))
  calculate :  #ClassMF4PDA        = MAX( 0, MIN( #ClassMF4PDA, #DistribAmt1 - #ClassAFPDA -
#ClassMF1PDA - #ClassMF2PDA - #ClassMF3PDA ) )
!
!
```

```
  calculate :  #ClassMF5PDA        = IF (#TrigEvent1 OR (#StepDown1 EQ 0.0)) _
                                THEN #DistribAmt1 - #ClassAFPDA - #ClassMF1PDA - #ClassMF2PDA -
#ClassMF3PDA - #ClassMF4PDA _
                                ELSE BBAL("AF5", "AF1", "AF2", "AF3", "AF4", "MF1", "MF2", "MF3", "MF4",
"MF5") - #ClassAFPDA - #ClassMF1PDA - #ClassMF2PDA - #ClassMF3PDA - #ClassMF4PDA _
                                - MIN(COLL_BAL(1) - #FloorOCTotal1, #MF5TargPct * COLL_BAL(1))
  calculate :  #ClassMF5PDA        = MAX( 0.0, MIN(BBAL("MF5"), #ClassMF5PDA ))
  calculate :  #ClassMF5PDA        = MAX( 0, MIN( #ClassMF5PDA, #DistribAmt1 - #ClassAFPDA -
#ClassMF1PDA - #ClassMF2PDA - #ClassMF3PDA - #ClassMF4PDA ) )
!
!
  calculate :  #ClassMF6PDA        = IF (#TrigEvent1 OR (#StepDown1 EQ 0.0)) _
                                THEN #DistribAmt1 - #ClassAFPDA - #ClassMF1PDA - #ClassMF2PDA -
#ClassMF3PDA - #ClassMF4PDA - #ClassMF5PDA _
                                ELSE BBAL("AF5", "AF1", "AF2", "AF3", "AF4", "MF1", "MF2", "MF3", "MF4",
"MF5", "MF6") - #ClassAFPDA - #ClassMF1PDA - #ClassMF2PDA - #ClassMF3PDA - #ClassMF4PDA -
#ClassMF5PDA _
                                - MIN(COLL_BAL(1) - #FloorOCTotal1, #MF6TargPct * COLL_BAL(1))
  calculate :  #ClassMF6PDA        = MAX( 0.0, MIN(BBAL("MF6"), #ClassMF6PDA ))
  calculate :  #ClassMF6PDA        = MAX( 0, MIN( #ClassMF6PDA, #DistribAmt1 - #ClassAFPDA -
#ClassMF1PDA - #ClassMF2PDA - #ClassMF3PDA - #ClassMF4PDA - #ClassMF5PDA ) )
!
!
  calculate :  #ClassAVPDA         = IF (#TrigEvent2 OR (#StepDown2 EQ 0.0)) _
                                THEN #DistribAmt2 _
                                ELSE BBAL("AV1", "AV2") _
                                - MIN(COLL_BAL(2) - #FloorOCTotal2, #AVTargPct * COLL_BAL(2))
  calculate :  #ClassAVPDA         = MAX( 0.0, MIN(BBAL("AV1", "AV2"), #ClassAVPDA ))
  calculate :  #ClassAVPDA         = MAX( 0, MIN( #ClassAVPDA, #DistribAmt2 ) )
!
!
  calculate :  #ClassMV1PDA        = IF (#TrigEvent2 OR (#StepDown2 EQ 0.0)) _
                                THEN #DistribAmt2 - #ClassAVPDA _
                                ELSE BBAL("AV1", "AV2", "MV1") - #ClassAVPDA _
                                - MIN(COLL_BAL(2) - #FloorOCTotal2, #MV1TargPct * COLL_BAL(2))
  calculate :  #ClassMV1PDA        = MAX( 0.0, MIN(BBAL("MV1"), #ClassMV1PDA ))
  calculate :  #ClassMV1PDA        = MAX( 0, MIN( #ClassMV1PDA, #DistribAmt2 - #ClassAVPDA ) )
!
!
  calculate :  #ClassMV2PDA        = IF (#TrigEvent2 OR (#StepDown2 EQ 0.0)) _
                                THEN #DistribAmt2 - #ClassAVPDA - #ClassMV1PDA _
                                ELSE BBAL("AV1", "AV2", "MV1", "MV2") - #ClassAVPDA - #ClassMV1PDA _
                                - MIN(COLL_BAL(2) - #FloorOCTotal2, #MV2TargPct * COLL_BAL(2))
  calculate :  #ClassMV2PDA        = MAX( 0.0, MIN(BBAL("MV2"), #ClassMV2PDA ))
  calculate :  #ClassMV2PDA        = MAX( 0, MIN( #ClassMV2PDA, #DistribAmt2 - #ClassAVPDA -
#ClassMV1PDA ) )
!
!
  calculate :  #ClassMV3PDA        = IF (#TrigEvent2 OR (#StepDown2 EQ 0.0)) _
                                THEN #DistribAmt2 - #ClassAVPDA - #ClassMV1PDA - #ClassMV2PDA _
                                ELSE BBAL("AV1", "AV2", "MV1", "MV2", "MV3") - #ClassAVPDA -
#ClassMV1PDA - #ClassMV2PDA _
                                - MIN(COLL_BAL(2) - #FloorOCTotal2, #MV3TargPct * COLL_BAL(2))
  calculate :  #ClassMV3PDA        = MAX( 0.0, MIN(BBAL("MV3"), #ClassMV3PDA ))
  calculate :  #ClassMV3PDA        = MAX( 0, MIN( #ClassMV3PDA, #DistribAmt2 - #ClassAVPDA -
```

```
#ClassMV1PDA - #ClassMV2PDA ) )
!
!
  calculate :  #ClassMV4PDA        = IF (#TrigEvent2 OR (#StepDown2 EQ 0.0)) _
                              THEN #DistribAmt2 - #ClassAVPDA - #ClassMV1PDA - #ClassMV2PDA -
#ClassMV3PDA _
                              ELSE BBAL("AV1", "AV2", "MV1", "MV2", "MV3", "MV4") - #ClassAVPDA -
#ClassMV1PDA - #ClassMV2PDA - #ClassMV3PDA _
                              - MIN(COLL_BAL(2) - #FloorOCTotal2, #MV4TargPct * COLL_BAL(2))
  calculate :  #ClassMV4PDA        = MAX( 0.0, MIN(BBAL("MV4"), #ClassMV4PDA ))
  calculate :  #ClassMV4PDA        = MAX( 0, MIN( #ClassMV4PDA, #DistribAmt2 - #ClassAVPDA -
#ClassMV1PDA - #ClassMV2PDA - #ClassMV3PDA ) )
!
!
  calculate :  #ClassMV5PDA        = IF (#TrigEvent2 OR (#StepDown2 EQ 0.0)) _
                              THEN #DistribAmt2 - #ClassAVPDA - #ClassMV1PDA - #ClassMV2PDA -
#ClassMV3PDA - #ClassMV4PDA _
                              ELSE BBAL("AV1", "AV2", "MV1", "MV2", "MV3", "MV4", "MV5") -
#ClassAVPDA - #ClassMV1PDA - #ClassMV2PDA - #ClassMV3PDA - #ClassMV4PDA _
                              - MIN(COLL_BAL(2) - #FloorOCTotal2, #MV5TargPct * COLL_BAL(2))
  calculate :  #ClassMV5PDA        = MAX( 0.0, MIN(BBAL("MV5"), #ClassMV5PDA ))
  calculate :  #ClassMV5PDA        = MAX( 0, MIN( #ClassMV5PDA, #DistribAmt2 - #ClassAVPDA -
#ClassMV1PDA - #ClassMV2PDA - #ClassMV3PDA - #ClassMV4PDA ) )
!
!
  calculate :  #ClassMV6PDA        = IF (#TrigEvent2 OR (#StepDown2 EQ 0.0)) _
                              THEN #DistribAmt2 - #ClassAVPDA - #ClassMV1PDA - #ClassMV2PDA -
#ClassMV3PDA - #ClassMV4PDA - #ClassMV5PDA _
                              ELSE BBAL("AV1", "AV2", "MV1", "MV2", "MV3", "MV4", "MV5", "MV6") -
#ClassAVPDA - #ClassMV1PDA - #ClassMV2PDA - #ClassMV3PDA - #ClassMV4PDA - #ClassMV5PDA
_
                              - MIN(COLL_BAL(2) - #FloorOCTotal2, #MV6TargPct * COLL_BAL(2))
  calculate :  #ClassMV6PDA        = MAX( 0.0, MIN(BBAL("MV6"), #ClassMV6PDA ))
  calculate :  #ClassMV6PDA        = MAX( 0, MIN( #ClassMV6PDA, #DistribAmt2 - #ClassAVPDA -
#ClassMV1PDA - #ClassMV2PDA - #ClassMV3PDA - #ClassMV4PDA - #ClassMV5PDA ) )
!
!
  calculate :  #PrincFrac2C        = ( COLL_P("2C_A") + COLL_P("2C_F") ) / COLL_P(2)
  calculate :  #PrincFrac2NC        = ( COLL_P("2NC_A") + COLL_P("2NC_F") ) / COLL_P(2)
!
  calculate :  #XtraPFrac2C        = ( COLL_P("2C_A") + COLL_P("2C_F") ) / COLL_P(2)
  calculate :  #XtraPFrac2NC        = ( COLL_P("2NC_A") + COLL_P("2NC_F") ) / COLL_P(2)
!
 calculate :  "AF" _
 NO_CHECK  CUSTOM GROUP 1   AMOUNT        = #ClassAFPDA
!
 calculate :  "MF1" _
 NO_CHECK  CUSTOM GROUP 1   AMOUNT        = #ClassMF1PDA
!
 calculate :  "MF2" _
 NO_CHECK  CUSTOM GROUP 1   AMOUNT        = #ClassMF2PDA
!
 calculate :  "MF3" _
 NO_CHECK  CUSTOM GROUP 1   AMOUNT        = #ClassMF3PDA
!
```

```
 calculate :  "MF4" _
NO_CHECK  CUSTOM GROUP 1   AMOUNT       = #ClassMF4PDA
!
 calculate :  "MF5" _
NO_CHECK  CUSTOM GROUP 1   AMOUNT       = #ClassMF5PDA
!
 calculate :  "MF6" _
NO_CHECK  CUSTOM GROUP 1   AMOUNT       = #ClassMF6PDA
!
 calculate :  "AV" _
NO_CHECK  CUSTOM GROUP 2   AMOUNT       = #ClassAVPDA
!
 calculate :  "MV1" _
NO_CHECK  CUSTOM GROUP 2   AMOUNT       = #ClassMV1PDA
!
 calculate :  "MV2" _
NO_CHECK  CUSTOM GROUP 2   AMOUNT       = #ClassMV2PDA
!
 calculate :  "MV3" _
NO_CHECK  CUSTOM GROUP 2   AMOUNT       = #ClassMV3PDA
!
 calculate :  "MV4" _
NO_CHECK  CUSTOM GROUP 2   AMOUNT       = #ClassMV4PDA
!
 calculate :  "MV5" _
NO_CHECK  CUSTOM GROUP 2   AMOUNT       = #ClassMV5PDA
!
 calculate :  "MV6" _
NO_CHECK  CUSTOM GROUP 2   AMOUNT       = #ClassMV6PDA
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!ALLOCATE PENALTIES TO YM RESERVE!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
------------------------------------
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
  subject to :  CEILING ( (COLL("PREPAYPENALTY",1) * 40%) )
       pay :  PREPAYPENALTY SEQUENTIAL("R_PP_1#1")
------------------------------------
!
------------------------------------
      from :  CASH_ACCOUNT (100, GROUP 1)
  subject to :  CEILING ( (COLL("PREPAYPENALTY",1) * (100% - 40%)) )
       pay :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
------------------------------------
!
------------------------------------
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
  subject to :  CEILING ( (COLL("PREPAYPENALTY",2) * 40%) )
       pay :  PREPAYPENALTY SEQUENTIAL("R_PP_2#1")
------------------------------------
!
------------------------------------
      from :  CASH_ACCOUNT (100, GROUP 2)
  subject to :  CEILING ( (COLL("PREPAYPENALTY",2) * (100% - 40%)) )
       pay :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
```

```
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!Initialize Variables for Wac Caps
----------------------------------- Initialize Variables for Wac Caps
            calculate : #Dummy1                 =           #NetWAC2C
            calculate : #Dummy2                 =           #AIO2C_RATE
            calculate : #Dummy3                 =           #YM_AV1
            calculate : #Dummy4                 =           #NetWAC2NC
            calculate : #Dummy5                 =           #AIO2NC_RATE
            calculate : #Dummy6                 =           #YM_AV2
            calculate : #Dummy7                 =           #NetWAC2
            calculate : #Dummy8                 =           #AIO2_RATE
            calculate : #Dummy9                 =           #YM_MV
            calculate : #Dummy10    =           #MVBAL
            calculate : #Dummy11    =           #YM_ALL
            calculate : #Dummy12    =           #ALLBAL
            calculate : #Dummy13    =           #AvlCapRate
            Calculate : #Dummy14    =           #G1BAL
-----------------------------------
!
----------------------------------- Pay Interest to AF  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "AF" )
-----------------------------------
 {#AF_Int}
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to AF  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "AF" )
-----------------------------------
 {#AF_InS}
-----------------------------------
!
----------------------------------- Pay Interest to MF1  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MF1" )
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to MF1  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF1" )
-----------------------------------
!
----------------------------------- Pay Interest to MF2  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MF2" )
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to MF2  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF2" )
-----------------------------------
!
```

```
---------------------------------- Pay Interest to MF3  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTEREST  SEQUENTIAL  ( "MF3" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MF3  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF3" )
----------------------------------
!
---------------------------------- Pay Interest to MF4  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTEREST  SEQUENTIAL  ( "MF4" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MF4  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF4" )
----------------------------------
!
---------------------------------- Pay Interest to MF5  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTEREST  SEQUENTIAL  ( "MF5" )

----------------------------------
!
---------------------------------- Pay Interest Shortfall to MF5  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF5" )
----------------------------------
!
---------------------------------- Pay Interest to MF6  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTEREST  SEQUENTIAL  ( "MF6" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MF6  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF6" )
----------------------------------
!
---------------------------------- Pay Principal to AF  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AF" )
----------------------------------
 {#AF_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF1  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF1" )
----------------------------------
 {#MF1_Prn}
----------------------------------
!
```

```
------------------------------------ Pay Principal to MF2  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF2" )
------------------------------------
 {#MF2_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF3  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF3" )
------------------------------------
 {#MF3_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF4  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF4" )
------------------------------------
 {#MF4_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF5  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF5" )
------------------------------------
 {#MF5_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF6  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF6" )
------------------------------------
 {#MF6_Prn}
------------------------------------
!
------------------------------------ Pay Principal to AF  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AF" )
------------------------------------
 {#AF_Prn}
------------------------------------
!
------------------------------------ Pay Principal to AF  From Fund YmRsvFnd1
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AF" )
------------------------------------
 {#AF_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF1  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF1" )
------------------------------------
 {#MF1_Prn}
------------------------------------
```

```
!
------------------------------------ Pay Principal to MF1  From Fund YmRsvFnd1
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF1" )
------------------------------------
 {#MF1_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF2  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF2" )
------------------------------------
 {#MF2_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF2  From Fund YmRsvFnd1
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF2" )
------------------------------------
 {#MF2_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF3  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF3" )
------------------------------------
 {#MF3_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF3  From Fund YmRsvFnd1
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF3" )
------------------------------------
 {#MF3_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF4  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF4" )
------------------------------------
 {#MF4_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF4  From Fund YmRsvFnd1
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF4" )
------------------------------------
 {#MF4_Prn}
------------------------------------
!
------------------------------------ Pay Principal to MF5  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF5" )
------------------------------------
 {#MF5_Prn}
```

```
-----------------------------------
!
----------------------------------- Pay Principal to MF5  From Fund YmRsvFnd1
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF5" )
-----------------------------------
 {#MF5_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MF6  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF6" )
-----------------------------------
 {#MF6_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MF6  From Fund YmRsvFnd1
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF6" )
-----------------------------------
 {#MF6_Prn}
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to AF  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "AF5"; "AF1"; "AF2"; "AF3"; "AF4" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF1  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF1" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF1  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF1" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF2  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF2  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF3  From Int
```

```
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF3" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF3  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF3" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF4  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF4" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF4  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF4" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF5  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF5" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF5  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF5" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF6  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF6" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF6  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF6" )
-----------------------------------
-----------------------------------
!
----------------------------------- Pay Interest to X  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "X" )
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to X  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "X" )
```

```
-----------------------------------
!
----------------------------------- Pay Interest to AV  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "AV" )
-----------------------------------
 {#AV_Int}
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to AV  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "AV" )
-----------------------------------
 {#AV_InS}
-----------------------------------
!
----------------------------------- Pay Interest to MV1  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MV1" )
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to MV1  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV1" )
-----------------------------------
!
----------------------------------- Pay Interest to MV2  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MV2" )
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to MV2  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV2" )
-----------------------------------
!
----------------------------------- Pay Interest to MV3  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MV3" )
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to MV3  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV3" )
-----------------------------------
!
----------------------------------- Pay Interest to MV4  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MV4" )
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to MV4  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV4" )
-----------------------------------
```

```
!
---------------------------------- Pay Interest to MV5  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS INTEREST  SEQUENTIAL  ( "MV5" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MV5  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV5" )
----------------------------------
!
---------------------------------- Pay Interest to MV6  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS INTEREST  SEQUENTIAL  ( "MV6" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MV6  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV6" )
----------------------------------
!
---------------------------------- Pay Principal to AV  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AV" )
----------------------------------
----------------------------------
!
---------------------------------- Pay Principal to MV1  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV1" )
----------------------------------
 {#MV1_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV2  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV2" )
----------------------------------
 {#MV2_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV3  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV3" )
----------------------------------
 {#MV3_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV4  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV4" )
----------------------------------
 {#MV4_Prn}
----------------------------------
!
```

```
------------------------------------ Pay Principal to MV5  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV5" )
-----------------------------------
 {#MV5_Prn}
-----------------------------------
!
------------------------------------ Pay Principal to MV6  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV6" )
-----------------------------------
 {#MV6_Prn}
-----------------------------------
!
------------------------------------ Pay Principal to AV  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AV" )
-----------------------------------
-----------------------------------
!
------------------------------------ Pay Principal to AV  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AV" )
-----------------------------------
 {#AV_Prn}{1}
-----------------------------------
!
------------------------------------ Pay Principal to MV1  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV1" )
-----------------------------------
 {#MV1_Prn}
-----------------------------------
!
------------------------------------ Pay Principal to MV1  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV1" )
-----------------------------------
 {#MV1_Prn}
-----------------------------------
!
------------------------------------ Pay Principal to MV2  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV2" )
-----------------------------------
 {#MV2_Prn}
-----------------------------------
!
------------------------------------ Pay Principal to MV2  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV2" )
-----------------------------------
 {#MV2_Prn}
-----------------------------------
!
```

```
---------------------------------- Pay Principal to MV3  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV3" )
----------------------------------
 {#MV3_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV3  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV3" )
----------------------------------
 {#MV3_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV4  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV4" )
----------------------------------
 {#MV4_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV4  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV4" )
----------------------------------
 {#MV4_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV5  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV5" )
----------------------------------
 {#MV5_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV5  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV5" )
----------------------------------
 {#MV5_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV6  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV6" )
----------------------------------
 {#MV6_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV6  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV6" )
----------------------------------
 {#MV6_Prn}
----------------------------------
```

```
-----------------------------------
!
!
----------------------------------- Payback Basis Risk Shortfall to AV  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "AV1"; "AV2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to AV  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "AV1"; "AV2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV1  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV1" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV1  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV1" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV2  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV2  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV3  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV3" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV3  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV3" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV4  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV4" )
-----------------------------------
```

```
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV4  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV4" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV5  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV5" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV5  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV5" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV6  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV6" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV6  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV6" )
-----------------------------------
!
----------------------------------- Pay Interest to X  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTEREST  SEQUENTIAL  ( "X" )
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to X  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTSHORT  SEQUENTIAL  ( "X" )
-----------------------------------
!
!
----------------------------------- Pay Interest to AV  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTEREST  SEQUENTIAL  ( "AV" )
-----------------------------------
 {#AV_Int}
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to AV  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTSHORT  SEQUENTIAL  ( "AV" )
-----------------------------------
 {#AV_InS}
-----------------------------------
```

```
!
---------------------------------- Pay Interest to MV1  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MV1" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MV1  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV1" )
----------------------------------
!
---------------------------------- Pay Interest to MV2  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MV2" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MV2  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV2" )
----------------------------------
!
---------------------------------- Pay Interest to MV3  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MV3" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MV3  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV3" )
----------------------------------
!
---------------------------------- Pay Interest to MV4  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MV4" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MV4  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV4" )
----------------------------------
!
---------------------------------- Pay Interest to MV5  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MV5" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MV5  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV5" )
----------------------------------
!
---------------------------------- Pay Interest to MV6  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MV6" )
----------------------------------
```

```
!
---------------------------------- Pay Interest Shortfall to MV6  From Int
      from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
       pay :  CLASS INTSHORT  SEQUENTIAL  ( "MV6" )
----------------------------------
!
---------------------------------- Pay Principal to AV  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AV" )
----------------------------------
----------------------------------
!
---------------------------------- Pay Principal to MV1  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV1" )
----------------------------------
 {#MV1_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV2  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV2" )
----------------------------------
 {#MV2_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV3  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV3" )
----------------------------------
 {#MV3_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV4  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV4" )
----------------------------------
 {#MV4_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV5  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV5" )
----------------------------------
 {#MV5_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MV6  From Princ
      from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV6" )
----------------------------------
 {#MV6_Prn}
----------------------------------
!
---------------------------------- Pay Principal to AV  From Int
```

```
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AV" )
-----------------------------------
-----------------------------------
!
----------------------------------- Pay Principal to AV  From Fund YmRsvFnd2
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AV" )
-----------------------------------
 {#AV_Prn}{0}
-----------------------------------
!
----------------------------------- Pay Principal to MV1  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV1" )
-----------------------------------
 {#MV1_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV1  From Fund YmRsvFnd1
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV1" )
-----------------------------------
 {#MV1_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV2  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV2" )
-----------------------------------
 {#MV2_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV2  From Fund YmRsvFnd1
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV2" )
-----------------------------------
 {#MV2_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV3  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV3" )
-----------------------------------
 {#MV3_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV3  From Fund YmRsvFnd1
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV3" )
-----------------------------------
 {#MV3_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV4  From Int
```

```
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV4" )
-----------------------------------
 {#MV4_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV4  From Fund YmRsvFnd1
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV4" )
-----------------------------------
 {#MV4_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV5  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV5" )
-----------------------------------
 {#MV5_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV5  From Fund YmRsvFnd1
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV5" )
-----------------------------------
 {#MV5_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV6  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV6" )
-----------------------------------
 {#MV6_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MV6  From Fund YmRsvFnd1
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MV6" )
-----------------------------------
 {#MV6_Prn}
-----------------------------------
!
-----------------------------------
----------------------------------- Payback Basis Risk Shortfall to AV  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "AV1"; "AV2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to AV  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "AV1"; "AV2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV1  From Int
```

```
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV1" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV1  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV1" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV2  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV2  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV3  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV3" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV3  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV3" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV4  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV4" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV4  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV4" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV5  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV5" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV5  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
```

```
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV5" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV6  From Int
     from :  SUBACCOUNT ( #Coll_I1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV6" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MV6  From Princ
     from :  SUBACCOUNT ( #Coll_P1, CLASS "GRP1")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MV6" )
-----------------------------------
-----------------------------------
!
----------------------------------- Pay Interest to AF  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "AF" )
-----------------------------------
 {#AF_Int}
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to AF  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "AF" )
-----------------------------------
 {#AF_InS}
-----------------------------------
!
----------------------------------- Pay Interest to MF1  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MF1" )
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to MF1  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF1" )
-----------------------------------
!
----------------------------------- Pay Interest to MF2  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MF2" )
-----------------------------------
!
----------------------------------- Pay Interest Shortfall to MF2  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF2" )
-----------------------------------
!
----------------------------------- Pay Interest to MF3  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MF3" )
-----------------------------------
!
```

```
---------------------------------- Pay Interest Shortfall to MF3  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF3" )
----------------------------------
!
---------------------------------- Pay Interest to MF4  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MF4" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MF4  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF4" )
----------------------------------
!
---------------------------------- Pay Interest to MF5  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MF5" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MF5  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF5" )
----------------------------------
!
---------------------------------- Pay Interest to MF6  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTEREST  SEQUENTIAL  ( "MF6" )
----------------------------------
!
---------------------------------- Pay Interest Shortfall to MF6  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS INTSHORT  SEQUENTIAL  ( "MF6" )
----------------------------------
!
---------------------------------- Pay Principal to AF  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AF" )
----------------------------------
 {#AF_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF1  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF1" )
----------------------------------
 {#MF1_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF2  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF2" )
----------------------------------
 {#MF2_Prn}
----------------------------------
!
```

```
---------------------------------- Pay Principal to MF3  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF3" )
----------------------------------
 {#MF3_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF4  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF4" )
----------------------------------
 {#MF4_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF5  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF5" )
----------------------------------
 {#MF5_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF6  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF6" )
----------------------------------
 {#MF6_Prn}
----------------------------------
!
---------------------------------- Pay Principal to AF  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AF" )
----------------------------------
 {#AF_Prn}
----------------------------------
!
---------------------------------- Pay Principal to AF  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "AF" )
----------------------------------
 {#AF_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF1  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF1" )
----------------------------------
 {#MF1_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF1  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF1" )
----------------------------------
 {#MF1_Prn}
----------------------------------
```

```
!
---------------------------------- Pay Principal to MF2  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF2" )
----------------------------------
 {#MF2_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF2  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF2" )
----------------------------------
 {#MF2_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF3  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF3" )
----------------------------------
 {#MF3_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF3  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF3" )
----------------------------------
 {#MF3_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF4  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF4" )
----------------------------------
 {#MF4_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF4  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF4" )
----------------------------------
 {#MF4_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF5  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF5" )
----------------------------------
 {#MF5_Prn}
----------------------------------
!
---------------------------------- Pay Principal to MF5  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF5" )
----------------------------------
 {#MF5_Prn}
```

```
-----------------------------------
!
----------------------------------- Pay Principal to MF6  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF6" )
-----------------------------------
 {#MF6_Prn}
-----------------------------------
!
----------------------------------- Pay Principal to MF6  From Fund YmRsvFnd2
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS PRINCIPAL SEQUENTIAL  ( "MF6" )
-----------------------------------
 {#MF6_Prn}
-----------------------------------
!
-----------------------------------
----------------------------------- Payback Basis Risk Shortfall to AF  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "AF5"; "AF1"; "AF2"; "AF3"; "AF4" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to AF  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "AF5"; "AF1"; "AF2"; "AF3"; "AF4" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF1  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF1" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF1  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF1" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF2  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF2  From Princ
      from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
       pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF2" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF3  From Int
      from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
```

```
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF3" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF3  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF3" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF4  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF4" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF4  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF4" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF5  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF5" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF5  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF5" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF6  From Int
     from :  SUBACCOUNT ( #Coll_I2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF6" )
-----------------------------------
-----------------------------------
!
----------------------------------- Payback Basis Risk Shortfall to MF6  From Princ
     from :  SUBACCOUNT ( #Coll_P2, CLASS "GRP2")
      pay :  CLASS COUPONCAP_SHORT PRO_RATA ( "MF6" )
-----------------------------------
-----------------------------------
     from :  HEDGE ("CAP12")
 subject to :  CEILING ((HEDGE ("CAP12", "OPTIMAL_PMT") - HEDGE ("CAP12", "ACTUAL_PMT")))
      pay :  CLASS COUPONCAP_SHORT PRO_RATA("AV1")
-----------------------------------
     from :  HEDGE ("CAP22")
 subject to :  CEILING ((HEDGE ("CAP22", "OPTIMAL_PMT") - HEDGE ("CAP22", "ACTUAL_PMT")))
      pay :  CLASS COUPONCAP_SHORT PRO_RATA("AV2")
-----------------------------------
     from :  HEDGE ("CAP32")
 subject to :  CEILING ((HEDGE ("CAP32", "OPTIMAL_PMT") - HEDGE ("CAP32", "ACTUAL_PMT")))
```

```
      pay :  CLASS COUPONCAP_SHORT PRO_RATA("MV1")
-----------------------------------
     from :  HEDGE ("CAP32")
 subject to :  CEILING ((HEDGE ("CAP32", "OPTIMAL_PMT") - HEDGE ("CAP32", "ACTUAL_PMT")))
      pay :  CLASS COUPONCAP_SHORT PRO_RATA("MV2")
-----------------------------------
     from :  HEDGE ("CAP32")
 subject to :  CEILING ((HEDGE ("CAP32", "OPTIMAL_PMT") - HEDGE ("CAP32", "ACTUAL_PMT")))
      pay :  CLASS COUPONCAP_SHORT PRO_RATA("MV3")
-----------------------------------
     from :  HEDGE ("CAP32")
 subject to :  CEILING ((HEDGE ("CAP32", "OPTIMAL_PMT") - HEDGE ("CAP32", "ACTUAL_PMT")))
      pay :  CLASS COUPONCAP_SHORT PRO_RATA("MV4")
-----------------------------------
     from :  HEDGE ("CAP32")
 subject to :  CEILING ((HEDGE ("CAP32", "OPTIMAL_PMT") - HEDGE ("CAP32", "ACTUAL_PMT")))
      pay :  CLASS COUPONCAP_SHORT PRO_RATA("MV5")
-----------------------------------
     from :  HEDGE ("CAP32")
 subject to :  CEILING ((HEDGE ("CAP32", "OPTIMAL_PMT") - HEDGE ("CAP32", "ACTUAL_PMT")))
      pay :  CLASS COUPONCAP_SHORT PRO_RATA("MV6")
-----------------------------------
     from :  HEDGE ("CAP12")
      pay :  AS_INTEREST ("R_2#1")
-----------------------------------
     from :  HEDGE ("CAP22")
      pay :  AS_INTEREST ("R_2#1")
-----------------------------------
     from :  HEDGE ("CAP32")
      pay :  AS_INTEREST ("R_2#1")
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! CALCULATE FULL TURBO !!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
-----------------------------------
  calculate :  #BeforeTurbo1    =
BBAL("AF5#1","AF1#1","AF2#1","AF3#1","AF4#1","MF1#1","MF2#1","MF3#1","MF4#1","MF5#1","MF6#1
")
  calculate :  #BeforeTurbo2    =
BBAL("AV1#1","AV2#1","MV1#1","MV2#1","MV3#1","MV4#1","MV5#1","MV6#1")
  calculate :  #CallTurbo1  = IF #CallTurbo EQ 0 THEN 0 ELSE ((COLL_BAL("LAGMON_3",1) /
#OrigCollBal1 ) < 20%)
  calculate :  #CallTurbo2  = IF #CallTurbo EQ 0 THEN 0 ELSE ((COLL_BAL("LAGMON_3",2) /
#OrigCollBal2 ) < 20%)
  calculate :  #YM1 = CREDIT_ENHANCEMENT ("YmRsvFnd1")
  calculate :  #YM2 = CREDIT_ENHANCEMENT ("YmRsvFnd2")

-----------------------------------
     when :  IS_TRUE ( #CallTurbo1 )
     from :  CLASS ("GRP1")
      pay :  CLASS BALANCE PRO_RATA ( "AF"; "MF1"; "MF2"; "MF3"; "MF4"; "MF5"; "MF6" )
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!      SENIOR PRINCIPAL 1      !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
  calculate :  #NasFracAF_T  = BBAL("AF5")/ BBAL("AF5", "AFNN")
  calculate :  #NasCeilAF_T = MIN ( BBAL("AF5"), CASH_ACCOUNT("AF"), #NasFracAF_T *
#NasShiftAF * CASH_ACCOUNT("AF") )
-----------------------------------
 subject to :  CEILING ( #NasCeilAF_T )
     from :  CLASS ( "AF" )
      pay :  CLASS BALANCE SEQUENTIAL ("AF5")
-----------------------------------
     from :  CLASS ( "AF" )
      pay :  CLASS BALANCE SEQUENTIAL ("AFNN", "AF5")
-----------------------------------
     from :  CLASS ( "AFNN" )
      pay :  CLASS BALANCE SEQUENTIAL ( "AF1", "AF2", "AF3", "AF4" )
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 {#AF_Prn}
 {#MF1_Prn}
 {#MF2_Prn}
 {#MF3_Prn}
 {#MF4_Prn}
 {#MF5_Prn}
 {#MF6_Prn}
-----------------------------------
-----------------------------------
     when :  IS_TRUE ( #CallTurbo1 )
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
      pay :  CLASS BALANCE PRO_RATA ( "AF"; "MF1"; "MF2"; "MF3"; "MF4"; "MF5"; "MF6" )
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!      SENIOR PRINCIPAL 1      !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  calculate :  #NasFracAF_T  = BBAL("AF5")/ BBAL("AF5", "AFNN")
  calculate :  #NasCeilAF_T = MIN ( BBAL("AF5"), CASH_ACCOUNT("AF"), #NasFracAF_T *
#NasShiftAF * CASH_ACCOUNT("AF") )
-----------------------------------
 subject to :  CEILING ( #NasCeilAF_T )
     from :  CLASS ( "AF" )
      pay :  CLASS BALANCE SEQUENTIAL ("AF5")
-----------------------------------
     from :  CLASS ( "AF" )
      pay :  CLASS BALANCE SEQUENTIAL ("AFNN", "AF5")
-----------------------------------
     from :  CLASS ( "AFNN" )
      pay :  CLASS BALANCE SEQUENTIAL ( "AF1", "AF2", "AF3", "AF4" )
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 {#AF_Prn}
 {#MF1_Prn}
 {#MF2_Prn}
 {#MF3_Prn}
 {#MF4_Prn}
 {#MF5_Prn}
 {#MF6_Prn}
-----------------------------------
     when :  IS_TRUE ( #CallTurbo2 )
```

```
      from :  CLASS ( "GRP2" )
       pay :  CLASS BALANCE PRO_RATA ( "AV"; "MV1"; "MV2"; "MV3"; "MV4"; "MV5"; "MV6" )
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!      SENIOR PRINCIPAL 2      !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  calculate :  #SeniorPrinc2_T       = CASH_ACCOUNT( "AV" )
  calculate :  #SeniorXtraP2_T       = 0
!
  calculate :  #PrincFrac2C_T        = ( COLL_P("2C_A") + COLL_P("2C_F") ) / COLL_P(2)
  calculate :  #PrincFrac2NC_T       = ( COLL_P("2NC_A") + COLL_P("2NC_F") ) / COLL_P(2)
!
  calculate :  #XtraPFrac2C_T        = ( COLL_P("2C_A") + COLL_P("2C_F") ) / COLL_P(2)
  calculate :  #XtraPFrac2NC_T       = ( COLL_P("2NC_A") + COLL_P("2NC_F") ) / COLL_P(2)
!
  calculate :  #SeniorPDA2C_T        = MIN( BBAL("AV1"), (#SeniorPrinc2_T * #PrincFrac2C_T) +
(#SeniorXtraP2_T * #XtraPFrac2C_T))
  calculate :  #SeniorPDA2NC_T       = MIN( BBAL("AV2"), (#SeniorPrinc2_T * #PrincFrac2NC_T) +
(#SeniorXtraP2_T * #XtraPFrac2NC_T))
!
-----------------------------------
      from :  SUBACCOUNT ( #SeniorPDA2C_T, CLASS "AV" )
       pay :  CLASS BALANCE SEQUENTIAL ( "AV1" )
-----------------------------------
      from :  SUBACCOUNT ( #SeniorPDA2NC_T, CLASS "AV" )
       pay :  CLASS BALANCE SEQUENTIAL ( "AV2" )
-----------------------------------
      from :  CLASS ( "AV" )
       pay :  CLASS BALANCE PRO_RATA ( "AV1"; "AV2" )
-----------------------------------
      from :  CLASS ( "AV1" )
       pay :  SEQUENTIAL ( "AV1#1" )
-----------------------------------
      from :  CLASS ( "AV2" )
       pay :  SEQUENTIAL ( "AV2#1" )
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 {#AV_Prn}{1}
 {#MV1_Prn}
 {#MV2_Prn}
 {#MV3_Prn}
 {#MV4_Prn}
 {#MV5_Prn}
 {#MV6_Prn}
-----------------------------------
      when :  IS_TRUE ( #CallTurbo2 )
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
       pay :  CLASS BALANCE PRO_RATA ( "AV"; "MV1"; "MV2"; "MV3"; "MV4"; "MV5"; "MV6" )
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!      SENIOR PRINCIPAL 2      !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  calculate :  #SeniorPrinc2_T       = CASH_ACCOUNT( "AV" )
  calculate :  #SeniorXtraP2_T       = 0
!
```

```
  calculate : #PrincFrac2C_T        = ( COLL_P("2C_A") + COLL_P("2C_F") ) / COLL_P(2)
  calculate : #PrincFrac2NC_T       = ( COLL_P("2NC_A") + COLL_P("2NC_F") ) / COLL_P(2)
!
  calculate : #XtraPFrac2C_T        = ( COLL_P("2C_A") + COLL_P("2C_F") ) / COLL_P(2)
  calculate : #XtraPFrac2NC_T       = ( COLL_P("2NC_A") + COLL_P("2NC_F") ) / COLL_P(2)
!
  calculate : #SeniorPDA2C_T        = MIN( BBAL("AV1"), (#SeniorPrinc2_T * #PrincFrac2C_T) +
(#SeniorXtraP2_T * #XtraPFrac2C_T))
  calculate : #SeniorPDA2NC_T       = MIN( BBAL("AV2"), (#SeniorPrinc2_T * #PrincFrac2NC_T) +
(#SeniorXtraP2_T * #XtraPFrac2NC_T))
!
------------------------------------
     from : SUBACCOUNT ( #SeniorPDA2C_T, CLASS "AV" )
      pay : CLASS BALANCE SEQUENTIAL ( "AV1" )
------------------------------------
     from : SUBACCOUNT ( #SeniorPDA2NC_T, CLASS "AV" )
      pay : CLASS BALANCE SEQUENTIAL ( "AV2" )
------------------------------------
     from : CLASS ( "AV" )
      pay : CLASS BALANCE PRO_RATA ( "AV1"; "AV2" )
------------------------------------
     from : CLASS ( "AV1" )
      pay : SEQUENTIAL ( "AV1#1" )
------------------------------------
     from : CLASS ( "AV2" )
      pay : SEQUENTIAL ( "AV2#1" )
------------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 {#AV_Prn}{1}
 {#MV1_Prn}
 {#MV2_Prn}
 {#MV3_Prn}
 {#MV4_Prn}
 {#MV5_Prn}
 {#MV6_Prn}
------------------------------------
!!CROSS TURBO AMOUNTS IF ONE GROUP PAYS OFF
!
------------------------------------
     when : IS_TRUE ( #CallTurbo1 AND BBAL("AV", "MV1", "MV2", "MV3", "MV4", "MV5", "MV6") LT
0.01)
     from : CLASS ("GRP2")
      pay : CLASS BALANCE PRO_RATA ( "AF"; "MF1"; "MF2"; "MF3"; "MF4"; "MF5"; "MF6" )
------------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!     SENIOR PRINCIPAL 1      !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!  calculate : #NasF_CROSS_TEST  = BBAL("AF5")/ BBAL("AF5", "AFNN")
  calculate : #NasF_CROSS_TEST  = BBAL("AF5")
  calculate : #NasFracAF_T  = BBAL("AF5")/ BBAL("AF5", "AFNN")
  calculate : #NasCeilAF_T = MIN ( BBAL("AF5"), CASH_ACCOUNT("AF"), #NasFracAF_T *
#NasShiftAF * CASH_ACCOUNT("AF") )
------------------------------------
 subject to : CEILING ( #NasCeilAF_T )
     from : CLASS ( "AF" )
```

```
      pay :  CLASS BALANCE SEQUENTIAL ("AF5")
-----------------------------------
     from :  CLASS ( "AF" )
      pay :  CLASS BALANCE SEQUENTIAL ("AFNN", "AF5")
-----------------------------------
     from :  CLASS ( "AFNN" )
      pay :  CLASS BALANCE SEQUENTIAL ( "AF1", "AF2", "AF3", "AF4" )
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 {#AF_Prn}
 {#MF1_Prn}
 {#MF2_Prn}
 {#MF3_Prn}
 {#MF4_Prn}
 {#MF5_Prn}
 {#MF6_Prn}
-----------------------------------
-----------------------------------
     when :  IS_TRUE ( #CallTurbo1 AND BBAL("AV", "MV1", "MV2", "MV3", "MV4", "MV5", "MV6") LT
0.01 )
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
      pay :  CLASS BALANCE PRO_RATA ( "AF"; "MF1"; "MF2"; "MF3"; "MF4"; "MF5"; "MF6" )
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!      SENIOR PRINCIPAL 1      !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  calculate :  #NasFracAF_T  = BBAL("AF5")/ BBAL("AF5", "AFNN")
  calculate :  #NasCeilAF_T = MIN ( BBAL("AF5"), CASH_ACCOUNT("AF"), #NasFracAF_T *
#NasShiftAF * CASH_ACCOUNT("AF") )
-----------------------------------
 subject to :  CEILING ( #NasCeilAF_T )
     from :  CLASS ( "AF" )
      pay :  CLASS BALANCE SEQUENTIAL ("AF5")
-----------------------------------
     from :  CLASS ( "AF" )
      pay :  CLASS BALANCE SEQUENTIAL ("AFNN", "AF5")
-----------------------------------
     from :  CLASS ( "AFNN" )
      pay :  CLASS BALANCE SEQUENTIAL ( "AF1", "AF2", "AF3", "AF4" )
-----------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 {#AF_Prn}
 {#MF1_Prn}
 {#MF2_Prn}

 {#MF3_Prn}
 {#MF4_Prn}
 {#MF5_Prn}
 {#MF6_Prn}
-----------------------------------
     when :  IS_TRUE ( #CallTurbo2 AND BBAL("AF", "MF1", "MF2", "MF3", "MF4", "MF5", "MF6") LT
0.01 )
     from :  CLASS ( "GRP1" )
      pay :  CLASS BALANCE PRO_RATA ( "AV"; "MV1"; "MV2"; "MV3"; "MV4"; "MV5"; "MV6" )
-----------------------------------
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!      SENIOR PRINCIPAL 2      !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  calculate :  #SeniorPrinc2_T       = CASH_ACCOUNT( "AV" )
  calculate :  #SeniorXtraP2_T       = 0
!
  calculate :  #PrincFrac2C_T        = ( COLL_P("2C_A") + COLL_P("2C_F") ) / COLL_P(2)
  calculate :  #PrincFrac2NC_T       = ( COLL_P("2NC_A") + COLL_P("2NC_F") ) / COLL_P(2)
!
  calculate :  #XtraPFrac2C_T        = ( COLL_P("2C_A") + COLL_P("2C_F") ) / COLL_P(2)
  calculate :  #XtraPFrac2NC_T       = ( COLL_P("2NC_A") + COLL_P("2NC_F") ) / COLL_P(2)
!
  calculate :  #SeniorPDA2C_T        = MIN( BBAL("AV1"), (#SeniorPrinc2_T * #PrincFrac2C_T) +
(#SeniorXtraP2_T * #XtraPFrac2C_T))
  calculate :  #SeniorPDA2NC_T       = MIN( BBAL("AV2"), (#SeniorPrinc2_T * #PrincFrac2NC_T) +
(#SeniorXtraP2_T * #XtraPFrac2NC_T))
!
------------------------------------
      from :  SUBACCOUNT ( #SeniorPDA2C_T, CLASS "AV" )
       pay :  CLASS BALANCE SEQUENTIAL ( "AV1" )
------------------------------------
      from :  SUBACCOUNT ( #SeniorPDA2NC_T, CLASS "AV" )
       pay :  CLASS BALANCE SEQUENTIAL ( "AV2" )
------------------------------------
      from :  CLASS ( "AV" )
       pay :  CLASS BALANCE PRO_RATA ( "AV1"; "AV2" )
------------------------------------
      from :  CLASS ( "AV1" )
       pay :  SEQUENTIAL ( "AV1#1" )
------------------------------------
      from :  CLASS ( "AV2" )
       pay :  SEQUENTIAL ( "AV2#1" )
------------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 {#AV_Prn}{1}
 {#MV1_Prn}
 {#MV2_Prn}
 {#MV3_Prn}
 {#MV4_Prn}
 {#MV5_Prn}
 {#MV6_Prn}
------------------------------------
      when :  IS_TRUE ( #CallTurbo2 AND BBAL("AF", "MF1", "MF2", "MF3", "MF4", "MF5", "MF6") LT
0.01 )
      from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
       pay :  CLASS BALANCE PRO_RATA ( "AV"; "MV1"; "MV2"; "MV3"; "MV4"; "MV5"; "MV6" )
------------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!      SENIOR PRINCIPAL 2      !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  calculate :  #SeniorPrinc2_T       = CASH_ACCOUNT( "AV" )
  calculate :  #SeniorXtraP2_T       = 0
!
  calculate :  #PrincFrac2C_T        = ( COLL_P("2C_A") + COLL_P("2C_F") ) / COLL_P(2)
  calculate :  #PrincFrac2NC_T       = ( COLL_P("2NC_A") + COLL_P("2NC_F") ) / COLL_P(2)
```

```
!
 calculate : #XtraPFrac2C_T      = ( COLL_P("2C_A") + COLL_P("2C_F") ) / COLL_P(2)
 calculate : #XtraPFrac2NC_T     = ( COLL_P("2NC_A") + COLL_P("2NC_F") ) / COLL_P(2)
!
 calculate : #SeniorPDA2C_T      = MIN( BBAL("AV1"), (#SeniorPrinc2_T * #PrincFrac2C_T) +
(#SeniorXtraP2_T * #XtraPFrac2C_T))
 calculate : #SeniorPDA2NC_T     = MIN( BBAL("AV2"), (#SeniorPrinc2_T * #PrincFrac2NC_T) +
(#SeniorXtraP2_T * #XtraPFrac2NC_T))
!
------------------------------------
     from :  SUBACCOUNT ( #SeniorPDA2C_T, CLASS "AV" )
      pay :  CLASS BALANCE SEQUENTIAL ( "AV1" )
------------------------------------
     from :  SUBACCOUNT ( #SeniorPDA2NC_T, CLASS "AV" )
      pay :  CLASS BALANCE SEQUENTIAL ( "AV2" )
------------------------------------
     from :  CLASS ( "AV" )
      pay :  CLASS BALANCE PRO_RATA ( "AV1"; "AV2" )
------------------------------------
     from :  CLASS ( "AV1" )
      pay :  SEQUENTIAL ( "AV1#1" )
------------------------------------
     from :  CLASS ( "AV2" )
      pay :  SEQUENTIAL ( "AV2#1" )
------------------------------------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 {#AV_Prn}{1}
 {#MV1_Prn}
 {#MV2_Prn}
 {#MV3_Prn}
 {#MV4_Prn}
 {#MV5_Prn}
 {#MV6_Prn}
------------------------------------
!
------------------------------------
 calculate : #AfterTurbo1    =
BBAL("AF5#1","AF1#1","AF2#1","AF3#1","AF4#1","MF1#1","MF2#1","MF3#1","MF4#1","MF5#1","MF6#1
")
 calculate : #AfterTurbo2    =
BBAL("AV1#1","AV2#1","MV1#1","MV2#1","MV3#1","MV4#1","MV5#1","MV6#1")
 calculate : #TurboPaid1     = #BeforeTurbo1 - #AfterTurbo1
 calculate : #TurboPaid2     = #BeforeTurbo1 - #AfterTurbo1
------------------------------------
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd1")
      pay :  AS_INTEREST ("R_1#1")
------------------------------------
     from :  CREDIT_ENHANCEMENT ("YmRsvFnd2")
      pay :  AS_INTEREST ("R_2#1")
------------------------------------
!!!!!!!! END OF TURBO !!!!!!!!!
------------------------------------
     from :  CLASS ( "GRP1" )
      pay :  AS_INTEREST ("R_1#1")
------------------------------------
```

```
      from :  CLASS ( "GRP2" )
       pay :  AS_INTEREST ("R_2#1")
-------------------------------------
  calculate : #BondBal1    =
BBAL("AF5#1","AF1#1","AF2#1","AF3#1","AF4#1","MF1#1","MF2#1","MF3#1","MF4#1","MF5#1","MF6#1
")
  calculate : #OC1        = MAX( 0, COLL_BAL(1) - #BondBal1 )
-------------------------------------
  calculate : #BondBal2    =
BBAL("AV1#1","AV2#1","MV1#1","MV2#1","MV3#1","MV4#1","MV5#1","MV6#1")
  calculate : #OC2        = MAX( 0, COLL_BAL(2) - #BondBal2 )
-------------------------------------
  calculate : #BondBal     = #BondBal1 + #BondBal2
-------------------------------------
-------------------------------------
 calculate: #CallBalGrp1 = COLL_BAL(1)
 calculate: #CallBalGrp2 = COLL_BAL(2)
-------------------------------------
--------------------- SECTION: "OPTR_GROUP_1"
-------------------------------------
      from :  CASH_ACCOUNT (100)
 subject to :  CEILING ( #CallBalGrp1)
       pay :  CLASS BALANCE SEQUENTIAL ( "GRP1" )
       pay :  CLASS MORE_INTEREST SEQUENTIAL ( "GRP1" )
-------------------------------------
      from :  CLASS ( "GRP1" )
       pay :  PRO_RATA ( "AF5#1"; "AF1#1"; "AF2#1"; "AF3#1"; "AF4#1" )
-------------------------------------
      from :  CLASS ( "GRP1" )
       pay :  PRO_RATA ( "MF1#1" )
-------------------------------------
      from :  CLASS ( "GRP1" )
       pay :  PRO_RATA ( "MF2#1" )
-------------------------------------
      from :  CLASS ( "GRP1" )
       pay :  PRO_RATA ( "MF3#1" )
-------------------------------------
      from :  CLASS ( "GRP1" )
       pay :  PRO_RATA ( "MF4#1" )
-------------------------------------
      from :  CLASS ( "GRP1" )
       pay :  PRO_RATA ( "MF5#1" )
-------------------------------------
      from :  CLASS ( "GRP1" )
       pay :  PRO_RATA ( "MF6#1" )
-------------------------------------
      from :  CLASS ( "GRP1" )
       pay :  AS_INTEREST ( "R_1#1" )
-------------------------------------
--------------------- SECTION: "OPTR_GROUP_2"
-------------------------------------
      from :  CASH_ACCOUNT (100)
 subject to :  CEILING ( #CallBalGrp2)
       pay :  CLASS BALANCE SEQUENTIAL ( "GRP2" )
       pay :  CLASS MORE_INTEREST SEQUENTIAL ( "GRP2" )
```

```
-----------------------------------
      from :  CLASS ( "GRP2" )
       pay :  PRO_RATA ( "AV1#1"; "AV2#1" )
-----------------------------------
      from :  CLASS ( "GRP2" )
       pay :  PRO_RATA ( "MV1#1" )
-----------------------------------
      from :  CLASS ( "GRP2" )
       pay :  PRO_RATA ( "MV2#1" )
-----------------------------------
      from :  CLASS ( "GRP2" )
       pay :  PRO_RATA ( "MV3#1" )
-----------------------------------
      from :  CLASS ( "GRP2" )
       pay :  PRO_RATA ( "MV4#1" )
-----------------------------------
      from :  CLASS ( "GRP2" )
       pay :  PRO_RATA ( "MV5#1" )
-----------------------------------
      from :  CLASS ( "GRP2" )
       pay :  PRO_RATA ( "MV6#1" )
-----------------------------------
      from :  CLASS ( "GRP2" )
       pay :  AS_INTEREST ( "R_2#1" )
-----------------------------------
Schedule "AIOCA" GROUP 2
DECLARE
VALUES OK
   20040825        58,874,000
   20040925        58,874,000
   20041025        58,874,000
   20041125        58,874,000
   20041225        58,874,000
   20050125        58,874,000
   20050225        58,874,000
   20050325        58,874,000
   20050425        58,874,000
   20050525        58,874,000
   20050625        58,874,000
   20050725        58,874,000
   20050825        58,874,000
   20050925        58,874,000
   20051025        58,874,000
   20051125        58,874,000
   20051225        58,874,000
   20060125        58,874,000
   20060225        58,874,000
   20060325        58,874,000
   20060425        58,874,000
   20060525        58,874,000
   20060625        58,874,000
   20060725              0
!
Schedule "AIOCF" GROUP 2
DECLARE
```

VALUES OK

20040825	10,376,000
20040925	10,376,000
20041025	10,376,000
20041125	10,376,000
20041225	10,376,000
20050125	10,376,000
20050225	10,376,000
20050325	10,376,000
20050425	10,376,000
20050525	10,376,000
20050625	10,376,000
20050725	10,376,000
20050825	10,376,000
20050925	10,376,000
20051025	10,376,000
20051125	10,376,000
20051225	10,376,000
20060125	10,376,000
20060225	10,376,000
20060325	10,376,000
20060425	10,376,000
20060525	10,376,000
20060625	10,376,000
20060725	0

!

Schedule "AIONCA" GROUP 2

DECLARE

VALUES OK

20040825	58,903,000
20040925	58,903,000
20041025	58,903,000
20041125	58,903,000
20041225	58,903,000
20050125	58,903,000
20050225	58,903,000
20050325	58,903,000
20050425	58,903,000
20050525	58,903,000
20050625	58,903,000
20050725	58,903,000
20050825	58,903,000
20050925	58,903,000
20051025	58,903,000
20051125	58,903,000
20051225	58,903,000
20060125	58,903,000
20060225	58,903,000
20060325	58,903,000
20060425	58,903,000
20060525	58,903,000
20060625	58,903,000
20060725	0

!

Schedule "AIONCF" GROUP 2

```
DECLARE
VALUES OK
   20040825        10,347,000
   20040925        10,347,000
   20041025        10,347,000
   20041125        10,347,000
   20041225        10,347,000
   20050125        10,347,000
   20050225        10,347,000
   20050325        10,347,000
   20050425        10,347,000
   20050525        10,347,000
   20050625        10,347,000
   20050725        10,347,000
   20050825        10,347,000
   20050925        10,347,000
   20051025        10,347,000
   20051125        10,347,000
   20051225        10,347,000
   20060125        10,347,000
   20060225        10,347,000
   20060325        10,347,000
   20060425        10,347,000
   20060525        10,347,000
   20060625        10,347,000
   20060725              0
!
Schedule "AIOCA_BAL" GROUP 2
DECLARE
VALUES OK
   20040825        58,874,000
   20040925        58,874,000
   20041025        58,874,000
   20041125        58,874,000
   20041225        58,874,000
   20050125        58,874,000
   20050225        58,874,000
   20050325        58,874,000
   20050425        58,874,000
   20050525        58,874,000
   20050625        58,874,000
   20050725        58,874,000
   20050825        58,874,000
   20050925        58,874,000
   20051025        58,874,000
   20051125        58,874,000
   20051225        58,874,000
   20060125        58,874,000
   20060225        58,874,000
   20060325        58,874,000
   20060425        58,874,000
   20060525        58,874,000
   20060625              0
!
Schedule "AIOCF_BAL" GROUP 2
```

```
DECLARE
VALUES OK
   20040825        10,376,000
   20040925        10,376,000
   20041025        10,376,000
   20041125        10,376,000
   20041225        10,376,000
   20050125        10,376,000
   20050225        10,376,000
   20050325        10,376,000
   20050425        10,376,000
   20050525        10,376,000
   20050625        10,376,000
   20050725        10,376,000
   20050825        10,376,000
   20050925        10,376,000
   20051025        10,376,000
   20051125        10,376,000
   20051225        10,376,000
   20060125        10,376,000
   20060225        10,376,000
   20060325        10,376,000
   20060425        10,376,000
   20060525        10,376,000
   20060625                0
!
Schedule "AIONCA_BAL" GROUP 2
DECLARE
VALUES OK
   20040825        58,903,000
   20040925        58,903,000
   20041025        58,903,000
   20041125        58,903,000
   20041225        58,903,000
   20050125        58,903,000
   20050225        58,903,000
   20050325        58,903,000
   20050425        58,903,000
   20050525        58,903,000
   20050625        58,903,000
   20050725        58,903,000
   20050825        58,903,000
   20050925        58,903,000
   20051025        58,903,000
   20051125        58,903,000
   20051225        58,903,000
   20060125        58,903,000
   20060225        58,903,000
   20060325        58,903,000
   20060425        58,903,000
   20060525        58,903,000
   20060625                0
!
Schedule "AIONCF_BAL" GROUP 2
DECLARE
```

```
VALUES OK
   20040825        10,347,000
   20040925        10,347,000
   20041025        10,347,000
   20041125        10,347,000
   20041225        10,347,000
   20050125        10,347,000
   20050225        10,347,000
   20050325        10,347,000
   20050425        10,347,000
   20050525        10,347,000
   20050625        10,347,000
   20050725        10,347,000
   20050825        10,347,000
   20050925        10,347,000
   20051025        10,347,000
   20051125        10,347,000
   20051225        10,347,000
   20060125        10,347,000
   20060225        10,347,000
   20060325        10,347,000
   20060425        10,347,000
   20060525        10,347,000
   20060625              0
!
!
!
 DEFINE DYNAMIC #YM_HE6MO80PCT = LOAN("SCHAM_PREP_AMT") * 80% *
LOAN("GROSSRATE") / 1200 * 6
!
!
 DEFINE TABLE "SRFTBLSAXSERV" (4, 2) = "PERIOD" "SRF"
    10.1      0.35
    30.1      0.45
    48.1      0.7
    999       0.85
!
 DEFINE DYNAMIC #SRFSAXSERV     = LOOKUP_TBL( "STEP", MONTHS_DIFF(CURDATE,
DEAL_FIRST_PAYDATE) + 1, "SRFTBLSAXSERV", "PERIOD", "SRF" )
!
 Collateral OVER
!
!     Factor     --Delay--
! Type  Date      P/Y   BV   Use BV for 0
 WL  20040701    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            WL   00   WAC          10.422 (     3605935.02 /     3605935.02 );     3605935.02
(#SRFSAXSERV + 0);       0.00          282:78    282:78      360 NO_CHECK
BALLOON SCHED_BOTH          180                    COMMERCIAL_OP   "USER_ResidName"
"ARM2_0"    ; GROUP "1F"                    PREPAY_FLAG NONE ;
```

M 2 WL 00 WAC 7.503 (85644469.64 / 85644469.64); 85644469.64
(#SRFSAXSERV + 0.02); 0.00 300:22 300:22 322 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "ARM2_0" ; GROUP "1F" PREPAY_FLAG
NONE ;
M 3 WL 00 WAC 7.259 (2335508.82 / 2335508.82); 2335508.82
(#SRFSAXSERV + 0.067); 0.00 303:2 303:2 305 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "ARM2_24" ; GROUP "1F"
PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 4 WL 00 WAC 6.657 (14807808.20 / 14807808.20); 14807808.20
(#SRFSAXSERV + 0.017); 0.00 307:2 307:2 309 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "ARM2_30" ; GROUP "1F"
PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 5 WL 00 WAC 6.769 (1954874.17 / 1954874.17); 1954874.17
(#SRFSAXSERV + 0); 0.00 331:3 331:3 334 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "ARM2_36" ; GROUP "1F"
PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 6 WL 00 WAC 6.341 (1436003.70 / 1436003.70); 1436003.70
(#SRFSAXSERV + 0.087); 0.00 359:1 359:1 360 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "ARM2_60" ; GROUP "1F"
PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 7 WL 00 WAC 6.51 (181559638.22 / 181559638.22); 181559638.22
(#SRFSAXSERV + 0.043); 0.00 332:1 332:1 333 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "ARM2_0" ; GROUP "1F" PREPAY_FLAG
YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 8 WL 00 WAC 7.247 (3828933.71 / 3828933.71); 3828933.71
(#SRFSAXSERV + 0); 0.00 354:3 354:3 357 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "ARM2_24" ; GROUP "1F"
PREPAY_FLAG YM FOR 60 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 9 WL 00 WAC 7 (150785.73 / 150785.73); 150785.73
(#SRFSAXSERV + 0); 0.00 179:1 179:1 180 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM2_30" ; GROUP "1F"
PREPAY_FLAG NONE ;
M 10 WL 00 WAC 6.514 (1192818.63 / 1192818.63); 1192818.63
(#SRFSAXSERV + 0.047); 0.00 178:2 178:2 180 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM2_36" ; GROUP "1F"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 11 WL 00 WAC 6.377 (1060850.82 / 1060850.82); 1060850.82
(#SRFSAXSERV + 0.193); 0.00 237:3 237:3 240 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM35_0" ; GROUP "1F"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 12 WL 00 WAC 6.774 (5321630.18 / 5321630.18); 5321630.18
(#SRFSAXSERV + 0.047); 0.00 358:2 358:2 360 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM35_0" ; GROUP "1F"
PREPAY_FLAG NONE ;
M 13 WL 00 WAC 7.721 (1022599.90 / 1022599.90); 1022599.90
(#SRFSAXSERV + 0); 0.00 359:1 359:1 360 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM35_24" ; GROUP "1F"
PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 14 WL 00 WAC 6.577 (2087002.62 / 2087002.62); 2087002.62
(#SRFSAXSERV + 0); 0.00 357:3 357:3 360 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM35_30" ; GROUP "1F"
PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 15 WL 00 WAC 6.319 (42049288.40 / 42049288.40); 42049288.40
(#SRFSAXSERV + 0.036); 0.00 358:2 358:2 360 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM35_36" ; GROUP "1F"

PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 16 WL 00 WAC 6.75 (543008.15 / 543008.15); 543008.15
(#SRFSAXSERV + 0); 0.00 356:4 356:4 360 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM35_36" ; GROUP "1F"
PREPAY_FLAG YM FOR 60 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 17 WL 00 WAC 7.21 (29944544.33 / 29944544.33); 29944544.33
(#SRFSAXSERV + 0.031); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.775 24 6 SYNC_INT 13.341 1.041 6.478 0 0 INIT_PERCAP
2.762 ORIG_GROSSRATE 7.21 COMMERCIAL_OP
"USER_ResidName" "ARM35_0" ; GROUP "2C_A" PREPAY_FLAG NONE ; TEASER
M 18 WL 00 WAC 6.5 (590117.47 / 590117.47); 590117.47
(#SRFSAXSERV + 0); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
5.694 23 6 SYNC_INT 12.811 1.155 6.385 0 0 INIT_PERCAP
2.075 ORIG_GROSSRATE 6.5 COMMERCIAL_OP
"USER_ResidName" "ARM35_24" ; GROUP "2C_A" PREPAY_FLAG YM FOR 6 NONE
; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 19 WL 00 WAC 8.075 (369265.05 / 369265.05); 369265.05
(#SRFSAXSERV + 0); 0.00 357:3 357:3 360 NO_CHECK ARM LIBOR_6MO
6.575 22 6 SYNC_INT 15.075 1.5 8.075 0 0
ORIG_GROSSRATE 8.075 COMMERCIAL_OP
"USER_ResidName" "ARM35_30" ; GROUP "2C_A" PREPAY_FLAG YM FOR 12
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 20 WL 00 WAC 6.739 (57754251.02 / 57754251.02); 57754251.02
(#SRFSAXSERV + 0.054); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.255 24 6 SYNC_INT 12.826 1.031 5.979 0 0 INIT_PERCAP
2.864 ORIG_GROSSRATE 6.739 COMMERCIAL_OP
"USER_ResidName" "ARM35_36" ; GROUP "2C_A" PREPAY_FLAG YM FOR 24
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 21 WL 00 WAC 7.115 (313661.03 / 313661.03); 313661.03
(#SRFSAXSERV + 0); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.75 24 6 SYNC_INT 13.115 1 6.115 0 0
ORIG_GROSSRATE 7.115 COMMERCIAL_OP
"USER_ResidName" "ARM35_0" ; GROUP "2C_A" PREPAY_FLAG YM FOR 30 NONE
; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 22 WL 00 WAC 7.477 (1713543.39 / 1713543.39); 1713543.39
(#SRFSAXSERV + 0); 0.00 357:3 357:3 360 NO_CHECK ARM LIBOR_6MO
5.997 22 6 SYNC_INT 13.975 1.249 6.716 0 0 INIT_PERCAP
2.209 ORIG_GROSSRATE 7.477 COMMERCIAL_OP
"USER_ResidName" "ARM35_12" ; GROUP "2C_A" PREPAY_FLAG YM FOR 36
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 23 WL 00 WAC 6.737 (19235566.27 / 19235566.27); 19235566.27
(#SRFSAXSERV + 0.029); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.619 24 6 SYNC_INT 12.737 1 5.806 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.737 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "ARM35_36" ; GROUP "2C_A" PREPAY_FLAG NONE ; TEASER
M 24 WL 00 WAC 6.111 (73728485.23 / 73728485.23); 73728485.23
(#SRFSAXSERV + 0.052); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.027 24 6 SYNC_INT 12.124 1.006 5.26 0 0 INIT_PERCAP 2.977
ORIG_GROSSRATE 6.111 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "ARM35_36" ; GROUP "2C_A" PREPAY_FLAG YM FOR 24
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 25 WL 00 WAC 6.072 (246148.74 / 246148.74); 246148.74
(#SRFSAXSERV + 0.239); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
5.374 23 6 SYNC_INT 12.072 1 5.072 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.072 AMORT NONE FOR 60 COMMERCIAL_OP

"USER_ResidName" "ARM_0" ; GROUP "2C_A" PREPAY_FLAG YM FOR 36 NONE ;
YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 26 WL 00 WAC 7.231 (14834176.13 / 14834176.13); 14834176.13
(#SRFSAXSERV + 0.019); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.989 36 6 SYNC_INT 13.346 1.042 6.469 0 0 INIT_PERCAP
2.873 ORIG_GROSSRATE 7.231 COMMERCIAL_OP
"USER_ResidName" "ARM_0" ; GROUP "2C_A" PREPAY_FLAG NONE ; TEASER
M 27 WL 00 WAC 7.616 (802311.49 / 802311.49); 802311.49
(#SRFSAXSERV + 0); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
5.903 35 6 SYNC_INT 14.168 1.276 7.616 0 0 INIT_PERCAP
1.736 ORIG_GROSSRATE 7.616 COMMERCIAL_OP
"USER_ResidName" "FXBall_0" ; GROUP "2C_A" PREPAY_FLAG YM FOR 6 NONE ;
YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 28 WL 00 WAC 6.494 (384222.35 / 384222.35); 384222.35
(#SRFSAXSERV + 0); 0.00 357:3 357:3 360 NO_CHECK ARM LIBOR_6MO
4.75 34 6 SYNC_INT 12.494 1 5.704 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.494 COMMERCIAL_OP
"USER_ResidName" "FXBall_36" ; GROUP "2C_A" PREPAY_FLAG YM FOR 24 NONE
; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 29 WL 00 WAC 6.886 (906017.80 / 906017.80); 906017.80
(#SRFSAXSERV + 0.147); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.997 36 6 SYNC_INT 12.886 1 5.997 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.886 COMMERCIAL_OP
"USER_ResidName" "FXBall_0" ; GROUP "2C_A" PREPAY_FLAG YM FOR 30 NONE ;
YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 30 WL 00 WAC 7.115 (27297411.70 / 27297411.70); 27297411.70
(#SRFSAXSERV + 0.039); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
5.555 35 6 SYNC_INT 13.377 1.106 6.512 0 0 INIT_PERCAP
2.592 ORIG_GROSSRATE 7.115 COMMERCIAL_OP
"USER_ResidName" "FX30_0" ; GROUP "2C_A" PREPAY_FLAG YM FOR 36 NONE ;
YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 31 WL 00 WAC 6.706 (5348929.55 / 5348929.55); 5348929.55
(#SRFSAXSERV + 0.017); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.588 36 6 SYNC_INT 12.706 1 5.845 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.706 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "FX30_12" ; GROUP "2C_A" PREPAY_FLAG NONE ; TEASER
M 32 WL 00 WAC 6.718 (416792.41 / 416792.41); 416792.41
(#SRFSAXSERV + 0); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
6.027 36 6 SYNC_INT 12.718 1 6.027 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.718 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "FX30_24" ; GROUP "2C_A" PREPAY_FLAG YM FOR 24 NONE
; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 33 WL 00 WAC 6.678 (832743.38 / 832743.38); 832743.38
(#SRFSAXSERV + 0.204); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
6.091 35 6 SYNC_INT 12.678 1 6.187 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.678 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "FX30_30" ; GROUP "2C_A" PREPAY_FLAG YM FOR 30 NONE
; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 34 WL 00 WAC 6.434 (14571745.39 / 14571745.39); 14571745.39
(#SRFSAXSERV + 0.063); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.397 36 6 SYNC_INT 12.46 1.008 5.63 0 0 INIT_PERCAP 2.956
ORIG_GROSSRATE 6.434 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "FX30_36" ; GROUP "2C_A" PREPAY_FLAG YM FOR 36 NONE
; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 35 WL 00 WAC 7.25 (397657.49 / 397657.49); 397657.49

(#SRFSAXSERV + 0); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
5 59 6 SYNC_INT 13.25 1 6.25 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 7.25 COMMERCIAL_OP
"USER_ResidName" "FX30_60" ; GROUP "2C_A" PREPAY_FLAG YM FOR 12 NONE
; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 36 WL 00 WAC 6.142 (808748.23 / 808748.23); 808748.23
(#SRFSAXSERV + 0.128); 0.00 357:3 357:3 360 NO_CHECK ARM LIBOR_6MO
5.025 58 6 SYNC_INT 12.142 1 5.387 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.142 COMMERCIAL_OP
"USER_ResidName" "FX525_0" ; GROUP "2C_A" PREPAY_FLAG YM FOR 36 NONE
; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 37 WL 00 WAC 7.526 (566938.65 / 566938.65); 566938.65
(#SRFSAXSERV + 0); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
6.38 61 6 SYNC_INT 13.526 1 7.104 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 7.526 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "FX525_36" ; GROUP "2C_A" PREPAY_FLAG YM FOR 36 NONE
; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 38 WL 00 WAC 9.125 (66579.36 / 66579.36); 66579.36
(#SRFSAXSERV + 0); 0.00 281:79 281:79 360 NO_CHECK ARM CMT_1YR
6.25 6 12 SYNC_INT 16.125 2 9.125 0 0
ORIG_GROSSRATE 9.125 COMMERCIAL_OP
"USER_ResidName" "FX525_36" ; GROUP "2C_A" PREPAY_FLAG NONE ;
M 39 WL 00 WAC 9.199 (2398475.26 / 2398475.26); 2398475.26
(#SRFSAXSERV + 0); 0.00 275:79 275:79 354 NO_CHECK ARM LIBOR_6MO
5.668 6 6 SYNC_INT 16.319 1.211 8.89 0 0 INIT_PERCAP 2.044
ORIG_GROSSRATE 9.199 COMMERCIAL_OP
"USER_ResidName" "FX525_0" ; GROUP "2C_A" PREPAY_FLAG NONE ; TEASER
M 40 WL 00 WAC 10.21 (365598.48 / 365598.48); 365598.48
(#SRFSAXSERV + 0); 0.00 281:79 281:79 360 NO_CHECK
BALLOON SCHED_BOTH 180 COMMERCIAL_OP "USER_ResidName"
"FX525_12" ; GROUP "2C_F" PREPAY_FLAG NONE ;
M 41 WL 00 WAC 7.5 (288999.30 / 288999.30); 288999.30
(#SRFSAXSERV + 0); 0.00 354:6 354:6 360 NO_CHECK
BALLOON SCHED_BOTH 180 COMMERCIAL_OP "USER_ResidName"
"FX525_24" ; GROUP "2C_F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 42 WL 00 WAC 7.395 (10252125.10 / 10252125.10); 10252125.10
(#SRFSAXSERV + 0.017); 0.00 301:19 301:19 320 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "FX525_36" ; GROUP "2C_F"
PREPAY_FLAG NONE ;
M 43 WL 00 WAC 6.67 (321762.99 / 321762.99); 321762.99
(#SRFSAXSERV + 0); 0.00 302:2 302:2 304 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "FX525_60" ; GROUP "2C_F"
PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 44 WL 00 WAC 6.592 (1337568.03 / 1337568.03); 1337568.03
(#SRFSAXSERV + 0.038); 0.00 313:2 313:2 315 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "ARM2_0" ; GROUP "2C_F"
PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 45 WL 00 WAC 6.562 (949254.46 / 949254.46); 949254.46
(#SRFSAXSERV + 0); 0.00 357:3 357:3 360 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "ARM2_0" ; GROUP "2C_F"
PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 46 WL 00 WAC 6.573 (25311902.06 / 25311902.06); 25311902.06
(#SRFSAXSERV + 0.032); 0.00 334:1 334:1 335 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "ARM2_24" ; GROUP "2C_F"

PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 47 WL 00 WAC 6.142 (605386.76 / 605386.76); 605386.76
(#SRFSAXSERV + 0); 0.00 358:2 358:2 360 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM2_30" ; GROUP "2C_F"
PREPAY_FLAG NONE ;
M 48 WL 00 WAC 6.66 (161556.14 / 161556.14); 161556.14
(#SRFSAXSERV + 0); 0.00 359:1 359:1 360 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM2_36" ; GROUP "2C_F"
PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 49 WL 00 WAC 6.508 (5075406.01 / 5075406.01); 5075406.01
(#SRFSAXSERV + 0.089); 0.00 358:2 358:2 360 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "ARM2_60" ; GROUP "2C_F"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 50 WL 00 WAC 7.39 (28428230.99 / 28428230.99); 28428230.99
(#SRFSAXSERV + 0.015); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.701 24 6 SYNC_INT 13.456 1.022 6.402 0 0 INIT_PERCAP
2.788 ORIG_GROSSRATE 7.39 COMMERCIAL_OP
"USER_ResidName" "ARM2_0" ; GROUP "2NC_A" PREPAY_FLAG NONE ; TEASER
M 51 WL 00 WAC 5.65 (224167.15 / 224167.15); 224167.15
(#SRFSAXSERV + 0.6); 0.00 357:3 357:3 360 NO_CHECK ARM LIBOR_6MO
4.4 22 6 SYNC_INT 12.65 1 4.4 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 5.65 COMMERCIAL_OP
"USER_ResidName" "ARM2_24" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 6
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 52 WL 00 WAC 7.128 (297578.28 / 297578.28); 297578.28
(#SRFSAXSERV + 0); 0.00 356:4 356:4 360 NO_CHECK ARM LIBOR_6MO
5.628 21 6 SYNC_INT 14.128 1.5 7.128 0 0
ORIG_GROSSRATE 7.128 COMMERCIAL_OP
"USER_ResidName" "ARM2_30" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 12
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 53 WL 00 WAC 6.581 (46756617.17 / 46756617.17); 46756617.17
(#SRFSAXSERV + 0.046); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
5.053 23 6 SYNC_INT 12.66 1.034 5.796 0 0 INIT_PERCAP 2.875
ORIG_GROSSRATE 6.581 COMMERCIAL_OP
"USER_ResidName" "ARM2_36" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 24
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 54 WL 00 WAC 9.5 (224383.53 / 224383.53); 224383.53
(#SRFSAXSERV + 0); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
6.75 25 6 SYNC_INT 15.5 1 8.5 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 9.5 COMMERCIAL_OP
"USER_ResidName" "ARM35_0" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 30
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 55 WL 00 WAC 6.703 (1103536.70 / 1103536.70); 1103536.70
(#SRFSAXSERV + 0); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
5.308 23 6 SYNC_INT 12.984 1.14 6.101 0 0 INIT_PERCAP 2.353
ORIG_GROSSRATE 6.703 COMMERCIAL_OP
"USER_ResidName" "ARM35_0" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 36
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 56 WL 00 WAC 7.303 (554897.04 / 554897.04); 554897.04
(#SRFSAXSERV + 0); 0.00 355:5 355:5 360 NO_CHECK ARM LIBOR_6MO
6.012 20 6 SYNC_INT 14.303 1.5 7.303 0 0
ORIG_GROSSRATE 7.303 COMMERCIAL_OP
"USER_ResidName" "ARM35_24" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 60
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 57 WL 00 WAC 7.25 (22808121.50 / 22808121.50); 22808121.50

(#SRFSAXSERV + 0.011); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
6.059 24 6 SYNC_INT 13.25 1 6.412 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 7.25 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "ARM35_30" ; GROUP "2NC_A" PREPAY_FLAG NONE ;
TEASER
M 58 WL 00 WAC 5.875 (439791.72 / 439791.72); 439791.72
(#SRFSAXSERV + 0); 0.00 356:4 356:4 360 NO_CHECK ARM LIBOR_6MO
4.375 21 6 SYNC_INT 12.875 1.5 5.875 0 0
ORIG_GROSSRATE 5.875 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "ARM35_36" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 12
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 59 WL 00 WAC 6.271 (89277233.57 / 89277233.57); 89277233.57
(#SRFSAXSERV + 0.038); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.079 24 6 SYNC_INT 12.288 1 5.392 0 0 INIT_PERCAP 2.984
ORIG_GROSSRATE 6.271 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "ARM35_36" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 24
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 60 WL 00 WAC 6.36 (214510.66 / 214510.66); 214510.66
(#SRFSAXSERV + 0); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.375 24 6 SYNC_INT 12.36 1 5.375 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.36 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "ARM35_0" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 30
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 61 WL 00 WAC 5.956 (2332242.44 / 2332242.44); 2332242.44
(#SRFSAXSERV + 0); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
4.751 23 6 SYNC_INT 12.196 1.12 5.134 0 0 INIT_PERCAP 2.639
ORIG_GROSSRATE 5.956 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "ARM35_24" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 36
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 62 WL 00 WAC 5.75 (439791.71 / 439791.71); 439791.71
(#SRFSAXSERV + 0); 0.00 298:2 298:2 300 NO_CHECK ARM LIBOR_6MO
4.75 35 6 SYNC_INT 12.75 1.5 5.75 0 0
ORIG_GROSSRATE 5.75 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "ARM35_30" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 36
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 63 WL 00 WAC 6.863 (13268812.23 / 13268812.23); 13268812.23
(#SRFSAXSERV + 0.04); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.538 36 6 SYNC_INT 12.965 1.051 6.098 0 0 INIT_PERCAP
2.784 ORIG_GROSSRATE 6.863 COMMERCIAL_OP
"USER_ResidName" "ARM35_36" ; GROUP "2NC_A" PREPAY_FLAG NONE ;
TEASER
M 64 WL 00 WAC 5.25 (420236.55 / 420236.55); 420236.55
(#SRFSAXSERV + 0); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
4.125 36 6 SYNC_INT 11.25 1 4.25 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 5.25 COMMERCIAL_OP
"USER_ResidName" "ARM35_0" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 24
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 65 WL 00 WAC 9.75 (169633.95 / 169633.95); 169633.95
(#SRFSAXSERV + 0); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
7.125 37 6 SYNC_INT 15.75 1 8.75 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 9.75 COMMERCIAL_OP
"USER_ResidName" "ARM35_12" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 30
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 66 WL 00 WAC 6.817 (21769305.17 / 21769305.17); 21769305.17
(#SRFSAXSERV + 0.035); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO

5.285 35 6 SYNC_INT 13.067 1.098 6.223 0 . 0 INIT_PERCAP
2.629 ORIG_GROSSRATE 6.817 COMMERCIAL_OP
"USER_ResidName" "ARM35_36" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 36
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 67 WL 00 WAC 6.787 (835828.66 / 835828.66); 835828.66
(#SRFSAXSERV + 0); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.166 36 6 SYNC_INT 13.787 1.245 6.787 0 0
ORIG_GROSSRATE 6.787 AMORT NONE FOR 36 COMMERCIAL_OP
"USER_ResidName" "ARM35_36" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 36
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 68 WL 00 WAC 7.036 (4448831.79 / 4448831.79); 4448831.79
(#SRFSAXSERV + 0.105); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.855 36 6 SYNC_INT 13.036 1 5.992 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 7.036 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "ARM_0" ; GROUP "2NC_A" PREPAY_FLAG NONE ; TEASER
M 69 WL 00 WAC 6.423 (14508248.19 / 14508248.19); 14508248.19
(#SRFSAXSERV + 0.04); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.335 36 6 SYNC_INT 12.431 1 5.641 0 0 INIT_PERCAP 2.942
ORIG_GROSSRATE 6.423 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "ARM_0" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 36
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 70 WL 00 WAC 6.625 (87354.75 / 87354.75); 87354.75
(#SRFSAXSERV + 0); 0.00 358:2 358:2 360 NO_CHECK ARM LIBOR_6MO
4.875 59 6 SYNC_INT 12.625 1 6.625 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.625 COMMERCIAL_OP
"USER_ResidName" "FXBall_0" ; GROUP "2NC_A" PREPAY_FLAG NONE ; TEASER
M 71 WL 00 WAC 7.638 (870596.29 / 870596.29); 870596.29
(#SRFSAXSERV + 0.161); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
6.53 60 6 SYNC_INT 13.638 1 6.913 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 7.638 COMMERCIAL_OP
"USER_ResidName" "FXBall_36" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 36
NONE ; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 72 WL 00 WAC 6.137 (1657304.62 / 1657304.62); 1657304.62
(#SRFSAXSERV + 0); 0.00 359:1 359:1 360 NO_CHECK ARM LIBOR_6MO
5.39 60 6 SYNC_INT 12.137 1 5.622 0 0 INIT_PERCAP 3
ORIG_GROSSRATE 6.137 AMORT NONE FOR 60 COMMERCIAL_OP
"USER_ResidName" "FXBall_0" ; GROUP "2NC_A" PREPAY_FLAG YM FOR 36 NONE
; YM_FORMULA (#YM_HE6MO80PCT); TEASER
M 73 WL 00 WAC 8.253 (148669.05 / 148669.05); 148669.05
(#SRFSAXSERV + 0); 0.00 282:78 282:78 360 NO_CHECK ARM CMT_1YR
6.281 7 12 SYNC_INT 15.866 2 7.755 0 0
ORIG_GROSSRATE 8.253 COMMERCIAL_OP
"USER_ResidName" "FXBall_0" ; GROUP "2NC_A" PREPAY_FLAG NONE ;
M 74 WL 00 WAC 9.658 (2124728.11 / 2124728.11); 2124728.11
(#SRFSAXSERV + 0); 0.00 280:80 280:80 360 NO_CHECK ARM LIBOR_6MO
5.74 5 6 SYNC_INT 16.631 1.071 9.658 0 0 INIT_PERCAP 2.808
ORIG_GROSSRATE 9.658 COMMERCIAL_OP
"USER_ResidName" "FX30_12" ; GROUP "2NC_A" PREPAY_FLAG NONE ; TEASER
M 75 WL 00 WAC 10.132 (401428.50 / 401428.50); 401428.50
(#SRFSAXSERV + 0); 0.00 281:79 281:79 360 NO_CHECK
BALLOON SCHED_BOTH 180 COMMERCIAL_OP "USER_ResidName"
"FX30_24" ; GROUP "2NC_F" PREPAY_FLAG NONE ;
M 76 WL 00 WAC 7.549 (9696008.31 / 9696008.31); 9696008.31
(#SRFSAXSERV + 0.027); 0.00 306:21 306:21 327 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "FX30_30" ; GROUP "2NC_F"

PREPAY_FLAG NONE ;
M 77 WL 00 WAC 7.681 (149220.25 / 149220.25); 149220.25
(#SRFSAXSERV + 0); 0.00 359:1 359:1 360 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "FX30_36" ; GROUP "2NC_F"
PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 78 WL 00 WAC 6.702 (2146988.14 / 2146988.14); 2146988.14
(#SRFSAXSERV + 0); 0.00 322:1 322:1 323 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "FX30_60" ; GROUP "2NC_F"
PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 79 WL 00 WAC 7.088 (785846.22 / 785846.22); 785846.22
(#SRFSAXSERV + 0); 0.00 339:1 339:1 340 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "FX525_0" ; GROUP "2NC_F"
PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 80 WL 00 WAC 6.533 (23494436.32 / 23494436.32); 23494436.32
(#SRFSAXSERV + 0.043); 0.00 334:2 334:2 336 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "FX525_36" ; GROUP "2NC_F"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 81 WL 00 WAC 6.908 (379794.11 / 379794.11); 379794.11
(#SRFSAXSERV + 0); 0.00 357:3 357:3 360 NO_CHECK
COMMERCIAL_OP "USER_ResidName" "FX525_36" ; GROUP "2NC_F"
PREPAY_FLAG YM FOR 60 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 82 WL 00 WAC 7.226 (369671.87 / 369671.87); 369671.87
(#SRFSAXSERV + 0); 0.00 179:1 179:1 180 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "FX525_0" ; GROUP "2NC_F"
PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 83 WL 00 WAC 5.875 (197603.36 / 197603.36); 197603.36
(#SRFSAXSERV + 0); 0.00 239:1 239:1 240 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "FX525_12" ; GROUP
"2NC_F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 84 WL 00 WAC 6.252 (1074123.97 / 1074123.97); 1074123.97
(#SRFSAXSERV + 0); 0.00 359:1 359:1 360 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "FX525_24" ; GROUP
"2NC_F" PREPAY_FLAG NONE ;
M 85 WL 00 WAC 6.232 (5857894.58 / 5857894.58); 5857894.58
(#SRFSAXSERV + 0); 0.00 358:2 358:2 360 NO_CHECK
AMORT NONE FOR 60 COMMERCIAL_OP "USER_ResidName" "FX525_36" ; GROUP
"2NC_F" PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (#YM_HE6MO80PCT);
M 86 WL 00 WAC 5.75 (237285.59 / 237285.59); 237285.59
(#SRFSAXSERV + 0); 0.00 357:3 357:3 360 NO_CHECK
AMORT NONE FOR 60
GROUP "2NC_F" PREPAY_FLAG YM FOR 60 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 1 WL 00 WAC 7.292 (14688229.21 / 14688229.21); 14688229.21
(#SRFSAXSERV + 0.028); 0.00 318:0 318:0 318 NO_CHECK
GROUP "1F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 2 WL 00 WAC 7.864 (548671.83 / 548671.83); 548671.83
(#SRFSAXSERV + 0.067); 0.00 305:0 305:0 305 NO_CHECK
GROUP "1F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 3 WL 00 WAC 7.262 (3478739.64 / 3478739.64); 3478739.64
(#SRFSAXSERV + 0.017); 0.00 309:0 309:0 309 NO_CHECK
GROUP "1F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 4 WL 00 WAC 7.374 (459250.84 / 459250.84); 459250.84
(#SRFSAXSERV + 0.000); 0.00 334:0 334:0 334 NO_CHECK

```
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT );
M      5          WL   00   WAC        6.946 (      337354.65 /      337354.65 );      337354.65
(#SRFSAXSERV + 0.087);   0.00        360:0    360:0      360 NO_CHECK
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (
#YM_HE6MO80PCT );
M      6          WL   00   WAC        7.115 (    42653085.68 /    42653085.68 );    42653085.68
(#SRFSAXSERV + 0.043);   0.00        333:0    333:0      333 NO_CHECK
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT );
M      7          WL   00   WAC        7.852 (      899516.22 /      899516.22 );      899516.22
(#SRFSAXSERV + 0.000);   0.00        357:0    357:0      357 NO_CHECK
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG YM FOR 60 NONE ; YM_FORMULA (
#YM_HE6MO80PCT );
M      8          WL   00   WAC        7.605 (       35423.49 /       35423.49 );       35423.49
(#SRFSAXSERV + 0.000);   0.00        180:0    180:0      180 NO_CHECK
AMORT NONE FOR        60
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG NONE ;
M      9          WL   00   WAC        7.119 (      280224.14 /      280224.14 );      280224.14
(#SRFSAXSERV + 0.047);   0.00        180:0    180:0      180 NO_CHECK
AMORT NONE FOR        60
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT );
M      10         WL   00   WAC        6.982 (      249221.47 /      249221.47 );      249221.47
(#SRFSAXSERV + 0.193);   0.00        240:0    240:0      240 NO_CHECK
AMORT NONE FOR        60
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT );
M      11         WL   00   WAC        7.379 (     1250189.47 /     1250189.47 );     1250189.47
(#SRFSAXSERV + 0.047);   0.00        360:0    360:0      360 NO_CHECK
AMORT NONE FOR        60
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG NONE ;
M      12         WL   00   WAC        8.326 (      240235.33 /      240235.33 );      240235.33
(#SRFSAXSERV + 0.000);   0.00        360:0    360:0      360 NO_CHECK
AMORT NONE FOR        60
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (
#YM_HE6MO80PCT );
M      13         WL   00   WAC        7.182 (      490291.24 /      490291.24 );      490291.24
(#SRFSAXSERV + 0.000);   0.00        360:0    360:0      360 NO_CHECK
AMORT NONE FOR        60
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT );
M      14         WL   00   WAC        6.924 (     9878472.46 /     9878472.46 );     9878472.46
(#SRFSAXSERV + 0.036);   0.00        360:0    360:0      360 NO_CHECK
AMORT NONE FOR        60
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT );
M      15         WL   00   WAC        7.355 (      127566.75 /      127566.75 );      127566.75
(#SRFSAXSERV + 0.000);   0.00        360:0    360:0      360 NO_CHECK
AMORT NONE FOR        60
GROUP "1F"   XPREFUND 2 at 0      PREPAY_FLAG YM FOR 60 NONE ; YM_FORMULA (
#YM_HE6MO80PCT );
M      16         WL   00   WAC        7.497 (     7034753.03 /     7034753.03 );     7034753.03
(#SRFSAXSERV + 0.031);   0.00        360:0    360:0      360 NO_CHECK ARM LIBOR_6MO
5.775   24   6 SYNC_INT        13.341        1.041        6.478        0      0  INIT_PERCAP
```

2.762 ORIG_GROSSRATE 7.497
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 17 WL 00 WAC 6.787 (138633.96 / 138633.96); 138633.96
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.694 23 6 SYNC_INT 12.811 1.155 6.385 0 0 INIT_PERCAP
2.075 ORIG_GROSSRATE 6.787
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 18 WL 00 WAC 8.362 (86749.97 / 86749.97); 86749.97
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
6.575 22 6 SYNC_INT 15.075 1.500 8.075 0 0
ORIG_GROSSRATE 8.362
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 19 WL 00 WAC 7.026 (13567977.01 / 13567977.01); 13567977.01
(#SRFSAXSERV + 0.054); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.255 24 6 SYNC_INT 12.826 1.031 5.979 0 0 INIT_PERCAP
2.864 ORIG_GROSSRATE 7.026
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 20 WL 00 WAC 7.402 (73687.14 / 73687.14); 73687.14
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.750 24 6 SYNC_INT 13.115 1.000 6.115 0 0
ORIG_GROSSRATE 7.402
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 21 WL 00 WAC 7.764 (402555.98 / 402555.98); 402555.98
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.997 22 6 SYNC_INT 13.975 1.249 6.716 0 0 INIT_PERCAP
2.209 ORIG_GROSSRATE 7.764
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 22 WL 00 WAC 7.024 (4518935.25 / 4518935.25); 4518935.25
(#SRFSAXSERV + 0.029); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.619 24 6 SYNC_INT 12.737 1.000 5.806 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 7.024 AMORT NONE FOR 60
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 23 WL 00 WAC 6.398 (17320740.50 / 17320740.50); 17320740.50
(#SRFSAXSERV + 0.052); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.027 24 6 SYNC_INT 12.124 1.006 5.260 0 0 INIT_PERCAP
2.977 ORIG_GROSSRATE 6.398 AMORT NONE FOR 60
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 24 WL 00 WAC 6.359 (57826.74 / 57826.74); 57826.74
(#SRFSAXSERV + 0.239); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.374 23 6 SYNC_INT 12.072 1.000 5.072 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 6.359 AMORT NONE FOR 60
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 25 WL 00 WAC 7.518 (3484934.19 / 3484934.19); 3484934.19
(#SRFSAXSERV + 0.019); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.989 36 6 SYNC_INT 13.346 1.042 6.469 0 0 INIT_PERCAP
2.873 ORIG_GROSSRATE 7.518
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 26 WL 00 WAC 7.903 (188483.86 / 188483.86); 188483.86

(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.903 35 6 SYNC_INT 14.168 1.276 7.616 0 0 INIT_PERCAP
1.736 ORIG_GROSSRATE 7.903
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 27 WL 00 WAC 6.781 (90263.83 / 90263.83); 90263.83
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
4.750 34 6 SYNC_INT 12.494 1.000 5.704 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 6.781
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 28 WL 00 WAC 7.173 (212847.17 / 212847.17); 212847.17
(#SRFSAXSERV + 0.147); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.997 36 6 SYNC_INT 12.886 1.000 5.997 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 7.173
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 29 WL 00 WAC 7.402 (6412872.63 / 6412872.63); 6412872.63
(#SRFSAXSERV + 0.039); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.555 35 6 SYNC_INT 13.377 1.106 6.512 0 0 INIT_PERCAP
2.592 ORIG_GROSSRATE 7.402
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 30 WL 00 WAC 6.993 (1256602.79 / 1256602.79); 1256602.79
(#SRFSAXSERV + 0.017); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.588 36 6 SYNC_INT 12.706 1.000 5.845 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 6.993 AMORT NONE FOR 60
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 31 WL 00 WAC 7.005 (97915.39 / 97915.39); 97915.39
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
6.027 36 6 SYNC_INT 12.718 1.000 6.027 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 7.005 AMORT NONE FOR 60
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 32 WL 00 WAC 6.965 (195633.10 / 195633.10); 195633.10
(#SRFSAXSERV + 0.204); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
6.091 35 6 SYNC_INT 12.678 1.000 6.187 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 6.965 AMORT NONE FOR 60
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 33 WL 00 WAC 6.721 (3423282.33 / 3423282.33); 3423282.33
(#SRFSAXSERV + 0.063); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.397 36 6 SYNC_INT 12.460 1.008 5.630 0 0 INIT_PERCAP
2.956 ORIG_GROSSRATE 6.721 AMORT NONE FOR 60
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 34 WL 00 WAC 7.537 (93420.10 / 93420.10); 93420.10
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.000 59 6 SYNC_INT 13.250 1.000 6.250 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 7.537
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 35 WL 00 WAC 6.429 (189996.01 / 189996.01); 189996.01
(#SRFSAXSERV + 0.128); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.025 58 6 SYNC_INT 12.142 1.000 5.387 0 0 INIT_PERCAP

3.000 ORIG_GROSSRATE 6.429
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 36 WL 00 WAC 7.813 (133188.64 / 133188.64); 133188.64
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
6.380 61 6 SYNC_INT 13.526 1.000 7.104 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 7.813 AMORT NONE FOR 60
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 37 WL 00 WAC 8.105 (67893.46 / 67893.46); 67893.46
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK
BALLOON SCHED_BOTH 180
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 38 WL 00 WAC 7.306 (1885928.58 / 1885928.58); 1885928.58
(#SRFSAXSERV + 0.022); 0.00 315:0 315:0 315 NO_CHECK
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 39 WL 00 WAC 7.275 (75590.51 / 75590.51); 75590.51
(#SRFSAXSERV + 0.000); 0.00 304:0 304:0 304 NO_CHECK
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 40 WL 00 WAC 7.197 (314229.56 / 314229.56); 314229.56
(#SRFSAXSERV + 0.038); 0.00 315:0 315:0 315 NO_CHECK
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 41 WL 00 WAC 7.167 (223004.59 / 223004.59); 223004.59
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 42 WL 00 WAC 7.178 (5946424.74 / 5946424.74); 5946424.74
(#SRFSAXSERV + 0.032); 0.00 335:0 335:0 335 NO_CHECK
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 43 WL 00 WAC 6.747 (142221.11 / 142221.11); 142221.11
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK
AMORT NONE FOR 60
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 44 WL 00 WAC 7.265 (37953.74 / 37953.74); 37953.74
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK
AMORT NONE FOR 60
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 45 WL 00 WAC 7.113 (1192345.02 / 1192345.02); 1192345.02
(#SRFSAXSERV + 0.089); 0.00 360:0 360:0 360 NO_CHECK
AMORT NONE FOR 60
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 46 WL 00 WAC 7.677 (6678531.52 / 6678531.52); 6678531.52
(#SRFSAXSERV + 0.015); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.701 24 6 SYNC_INT 13.456 1.022 6.402 0 0 INIT_PERCAP
2.788 ORIG_GROSSRATE 7.677
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 47 WL 00 WAC 5.937 (52662.70 / 52662.70); 52662.70
(#SRFSAXSERV + 0.600); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
4.400 22 6 SYNC_INT 12.650 1.000 4.400 0 0 INIT_PERCAP

3.000 ORIG_GROSSRATE 5.937
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 48 WL 00 WAC 7.415 (69908.89 / 69908.89); 69908.89
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.628 21 6 SYNC_INT 14.128 1.500 7.128 0 0
ORIG_GROSSRATE 7.415
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 49 WL 00 WAC 6.868 (10984346.61 / 10984346.61); 10984346.61
(#SRFSAXSERV + 0.046); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.053 23 6 SYNC_INT 12.660 1.034 5.796 0 0 INIT_PERCAP
2.875 ORIG_GROSSRATE 6.868
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 50 WL 00 WAC 9.787 (52713.53 / 52713.53); 52713.53
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
6.750 25 6 SYNC_INT 15.500 1.000 8.500 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 9.787
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 51 WL 00 WAC 6.990 (259249.52 / 259249.52); 259249.52
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.308 23 6 SYNC_INT 12.984 1.140 6.101 0 0 INIT_PERCAP
2.353 ORIG_GROSSRATE 6.990
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 52 WL 00 WAC 7.590 (130359.77 / 130359.77); 130359.77
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
6.012 20 6 SYNC_INT 14.303 1.500 7.303 0 0
ORIG_GROSSRATE 7.590
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 60 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 53 WL 00 WAC 7.537 (5358221.52 / 5358221.52); 5358221.52
(#SRFSAXSERV + 0.011); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
6.059 24 6 SYNC_INT 13.250 1.000 6.412 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 7.537 AMORT NONE FOR 60
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 54 WL 00 WAC 6.162 (103318.53 / 103318.53); 103318.53
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
4.375 21 6 SYNC_INT 12.875 1.500 5.875 0 0
ORIG_GROSSRATE 6.162 AMORT NONE FOR 60
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 55 WL 00 WAC 6.558 (20973546.30 / 20973546.30); 20973546.30
(#SRFSAXSERV + 0.038); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.079 24 6 SYNC_INT 12.288 1.000 5.392 0 0 INIT_PERCAP
2.984 ORIG_GROSSRATE 6.558 AMORT NONE FOR 60
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 56 WL 00 WAC 6.647 (50394.14 / 50394.14); 50394.14
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.375 24 6 SYNC_INT 12.360 1.000 5.375 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 6.647 AMORT NONE FOR 60
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (

#YM_HE6MO80PCT); TEASER
M 57 WL 00 WAC 6.243 (547904.47 / 547904.47); 547904.47
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
4.751 23 6 SYNC_INT 12.196 1.120 5.134 0 0 INIT_PERCAP
2.639 ORIG_GROSSRATE 6.243 AMORT NONE FOR 60
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 58 WL 00 WAC 6.037 (103318.52 / 103318.52); 103318.52
(#SRFSAXSERV + 0.000); 0.00 300:0 300:0 300 NO_CHECK ARM LIBOR_6MO
4.750 35 6 SYNC_INT 12.750 1.500 5.750 0 0
ORIG_GROSSRATE 6.037 AMORT NONE FOR 60
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 59 WL 00 WAC 7.150 (3117189.41 / 3117189.41); 3117189.41
(#SRFSAXSERV + 0.040); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.538 36 6 SYNC_INT 12.965 1.051 6.098 0 0 INIT_PERCAP
2.784 ORIG_GROSSRATE 7.150
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 60 WL 00 WAC 5.537 (98724.51 / 98724.51); 98724.51
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
4.125 36 6 SYNC_INT 11.250 1.000 4.250 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 5.537
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 61 WL 00 WAC 10.037 (39851.43 / 39851.43); 39851.43
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
7.125 37 6 SYNC_INT 15.750 1.000 8.750 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 10.037
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 30 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 62 WL 00 WAC 7.104 (5114176.52 / 5114176.52); 5114176.52
(#SRFSAXSERV + 0.035); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.285 35 6 SYNC_INT 13.067 1.098 6.223 0 0 INIT_PERCAP
2.629 ORIG_GROSSRATE 7.104
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 63 WL 00 WAC 7.074 (196357.91 / 196357.91); 196357.91
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.166 36 6 SYNC_INT 13.787 1.245 6.787 0 0
ORIG_GROSSRATE 7.074 AMORT NONE FOR 36
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 64 WL 00 WAC 7.323 (1045146.41 / 1045146.41); 1045146.41
(#SRFSAXSERV + 0.105); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.855 36 6 SYNC_INT 13.036 1.000 5.992 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 7.323 AMORT NONE FOR 60
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 65 WL 00 WAC 6.710 (3408365.18 / 3408365.18); 3408365.18
(#SRFSAXSERV + 0.040); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.335 36 6 SYNC_INT 12.431 1.000 5.641 0 0 INIT_PERCAP
2.942 ORIG_GROSSRATE 6.710 AMORT NONE FOR 60
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 66 WL 00 WAC 6.912 (20521.91 / 20521.91); 20521.91
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO

4.875 59 6 SYNC_INT 12.625 1.000 6.625 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 6.912
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 67 WL 00 WAC 7.925 (204525.74 / 204525.74); 204525.74
(#SRFSAXSERV + 0.161); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
6.530 60 6 SYNC_INT 13.638 1.000 6.913 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 7.925
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 68 WL 00 WAC 6.424 (389343.99 / 389343.99); 389343.99
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK ARM LIBOR_6MO
5.390 60 6 SYNC_INT 12.137 1.000 5.622 0 0 INIT_PERCAP
3.000 ORIG_GROSSRATE 6.424 AMORT NONE FOR 60
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT); TEASER
M 69 WL 00 WAC 7.418 (1709182.19 / 1709182.19); 1709182.19
(#SRFSAXSERV + 0.037); 0.00 326:0 326:0 326 NO_CHECK
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 70 WL 00 WAC 8.286 (35055.73 / 35055.73); 35055.73
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 6 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 71 WL 00 WAC 7.307 (504383.42 / 504383.42); 504383.42
(#SRFSAXSERV + 0.000); 0.00 323:0 323:0 323 NO_CHECK
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 12 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 72 WL 00 WAC 7.693 (184615.73 / 184615.73); 184615.73
(#SRFSAXSERV + 0.000); 0.00 340:0 340:0 340 NO_CHECK
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 24 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 73 WL 00 WAC 7.138 (5519454.73 / 5519454.73); 5519454.73
(#SRFSAXSERV + 0.043); 0.00 336:0 336:0 336 NO_CHECK
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 74 WL 00 WAC 7.513 (89223.53 / 89223.53); 89223.53
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 60 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 75 WL 00 WAC 7.831 (86845.55 / 86845.55); 86845.55
(#SRFSAXSERV + 0.000); 0.00 180:0 180:0 180 NO_CHECK
AMORT NONE FOR 60
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 76 WL 00 WAC 6.480 (46422.17 / 46422.17); 46422.17
(#SRFSAXSERV + 0.000); 0.00 240:0 240:0 240 NO_CHECK
AMORT NONE FOR 60
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 77 WL 00 WAC 6.857 (252339.68 / 252339.68); 252339.68
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK
AMORT NONE FOR 60
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 78 WL 00 WAC 6.837 (1376171.94 / 1376171.94); 1376171.94
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK
AMORT NONE FOR 60

GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 36 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 79 WL 00 WAC 6.355 (55744.56 / 55744.56); 55744.56
(#SRFSAXSERV + 0.000); 0.00 360:0 360:0 360 NO_CHECK
AMORT NONE FOR 60
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG YM FOR 60 NONE ; YM_FORMULA (
#YM_HE6MO80PCT);
M 80 WL 00 WAC 9.394 (11299434.55 / 11299434.55); 11299434.55
(#SRFSAXSERV + 0.000); 0.00 291:69 291:69 360 NO_CHECK
BALLOON SCHED_BOTH 180
GROUP "1F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 81 WL 00 WAC 9.271 (21413698.24 / 21413698.24); 21413698.24
(#SRFSAXSERV + 0.000); 0.00 245:69 245:69 314 NO_CHECK
GROUP "1F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 82 WL 00 WAC 8.248 (617418.87 / 617418.87); 617418.87
(#SRFSAXSERV + 0.000); 0.00 289:71 289:71 360 NO_CHECK ARM CMT_1YR
6.177 9 12 SYNC_INT 15.699 2.000 8.183 0 0
ORIG_GROSSRATE 8.248
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 83 WL 00 WAC 9.366 (2736361.83 / 2736361.83); 2736361.83
(#SRFSAXSERV + 0.000); 0.00 290:70 290:70 360 NO_CHECK ARM LIBOR_6MO
6.128 4 6 SYNC_INT 16.336 1.192 9.207 0 0 INIT_PERCAP 2.609
ORIG_GROSSRATE 9.366
GROUP "2C_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 84 WL 00 WAC 9.330 (458201.23 / 458201.23); 458201.23
(#SRFSAXSERV + 0.000); 0.00 291:69 291:69 360 NO_CHECK
BALLOON SCHED_BOTH 180
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 85 WL 00 WAC 9.123 (656113.64 / 656113.64); 656113.64
(#SRFSAXSERV + 0.000); 0.00 232:69 232:69 301 NO_CHECK
GROUP "2C_F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 172 WL 00 WAC 9.250 (63866.87 / 63866.87); 63866.87
(#SRFSAXSERV + 0.000); 0.00 291:69 291:69 360 NO_CHECK ARM CMT_1YR
6.875 4 12 SYNC_INT 16.250 2.000 9.250 0 0
ORIG_GROSSRATE 9.250
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 173 WL 00 WAC 9.539 (3551031.10 / 3551031.10); 3551031.10
(#SRFSAXSERV + 0.000); 0.00 289:71 289:71 360 NO_CHECK ARM LIBOR_6MO
6.246 4 6 SYNC_INT 16.368 1.058 9.313 0 0 INIT_PERCAP 2.531
ORIG_GROSSRATE 9.539
GROUP "2NC_A" XPREFUND 2 at 0 PREPAY_FLAG NONE ; TEASER
M 174 WL 00 WAC 9.500 (35964.44 / 35964.44); 35964.44
(#SRFSAXSERV + 0.000); 0.00 290:70 290:70 360 NO_CHECK
BALLOON SCHED_BOTH 180
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;
M 175 WL 00 WAC 9.378 (826427.62 / 826427.62); 826427.62
(#SRFSAXSERV + 0.000); 0.00 245:69 245:69 314 NO_CHECK
GROUP "2NC_F" XPREFUND 2 at 0 PREPAY_FLAG NONE ;