



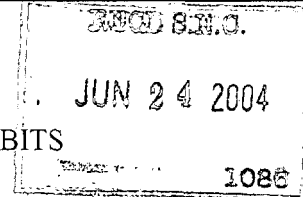
04033859

UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
Washington, D. C. 20549

OMB APPROVAL
OMB Number: 3235-0327
Expires: July 31, 2004
Estimated average burden hours per response... 0.10

FORM SE

FORM FOR SUBMISSION OF PAPER FORMAT EXHIBITS
BY ELECTRONIC FILERS



Banc of America Mortgage Securities, Inc.
Exact name of registrant as specified in charter

0001207409
Registrant CIK Number

8-K FOR 6/25/04
Electronic report, schedule or registration statement
of which the documents are a part (give period of report)

333-105940
SEC file number, if available

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

PROCESSED

JUN 25 2004

THOMSON
FINANCIAL

SIGNATURES

Filings Made By the Registrant:

The Registrant has duly caused this form to be signed on its behalf by the undersigned, thereunto
duly authorized, in the City of Alpharetta, State of Georgia.

Banc of America Mortgage Securities, Inc.
(Registrant)

By: Judy Lowman
Name: Judy Lowman
Title: Vice President

Filings Made by Person Other Than the Registrant:

After reasonable inquiry and to the best of my knowledge and belief, I certify on _____,
2003, that the information set forth in this statement is true and complete.

By: _____
(Name)

(Title)

BOAMS 2004-F-Marketing - Price/Yield - 1A1

Balance \$222,600,000.00 Delay 24
Initial Coupon 3.756763827 Dated 6/1/2004
Settle 6/28/2004 First Payment 7/25/2004

Price	5 CPR To Roll Yield	10 CPR To Roll Yield	15 CPR To Roll Yield	20 CPR To Roll Yield	25 CPR To Roll Yield	30 CPR To Roll Yield	40 CPR To Roll Yield	50 CPR To Roll Yield
98-20.00	4.248	4.283	4.323	4.367	4.416	4.471	4.601	4.767
98-21.00	4.235	4.269	4.308	4.351	4.398	4.451	4.578	4.739
98-22.00	4.222	4.255	4.293	4.334	4.381	4.432	4.555	4.712
98-23.00	4.209	4.241	4.278	4.318	4.363	4.413	4.532	4.684
98-24.00	4.196	4.228	4.263	4.302	4.345	4.394	4.509	4.657
98-25.00	4.183	4.214	4.248	4.286	4.328	4.375	4.486	4.629
98-26.00	4.170	4.200	4.233	4.269	4.310	4.356	4.464	4.602
98-27.00	4.157	4.186	4.218	4.253	4.293	4.337	4.441	4.574
98-28.00	4.144	4.172	4.203	4.237	4.275	4.317	4.418	4.547
98-29.00	4.132	4.158	4.188	4.221	4.257	4.298	4.395	4.520
98-30.00	4.119	4.144	4.173	4.205	4.240	4.279	4.373	4.492
WAL	2.634	2.442	2.260	2.085	1.920	1.764	1.478	1.223
Mod Durm	2.450	2.273	2.105	1.944	1.792	1.648	1.384	1.148
Principal Window Begin	07/25/2004	07/25/2004	07/25/2004	07/25/2004	07/25/2004	07/25/2004	07/25/2004	07/25/2004
Principal Window End	05/25/2007	05/25/2007	05/25/2007	05/25/2007	05/25/2007	05/25/2007	05/25/2007	05/25/2007
LIBOR_1YR	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2

This Structural Term Sheet, Collateral Term Sheet, or Computational Materials, as appropriate (the "material"), is for your private information and Banc of America Securities LLC (the "Underwriter") is not soliciting any action based upon it. This material is not to be construed as an offer to sell or the solicitation of any offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. This material is based on information that the Underwriter considers reliable, but the Underwriter does not represent that it is accurate or complete and it should not be relied upon as such. By accepting this material the recipient agrees that it will not distribute or provide the material to any other person. The information contained in this material may pertain to securities that ultimately are not sold. The information contained in this material may be based on assumptions regarding market conditions and other matters as reflected herein. The Underwriter makes no representation regarding the reasonableness of such assumptions or the likelihood that any of such assumptions will coincide with actual market conditions or events, and this material should not be relied upon for such purposes. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material may, from time to time, have long or short positions in, and buy and sell, the securities mentioned herein or derivatives thereof (including options). This material may be filed with the Securities and Exchange Commission (the "SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on this material only. Information in this material regarding any assets backing any securities discussed herein supersedes all prior information regarding such assets. Any information in the material, whether regarding the assets backing any securities discussed herein or otherwise, will be superseded by the information contained in any final prospectus for any securities actually sold to you. This material is furnished solely by the Underwriter and not by the issuer of the securities. The issuer of the securities has not prepared, reviewed or participated in the preparation of this material, is not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as underwriter and not acting as agent for the issuer in connection with the proposed transaction.

Balance	\$272,600,000.00	Delay	24
Initial Coupon	3.756763827	Dated	6/1/2004
Settle	6/28/2004	First Payment	7/25/2004

[illegible]

BOAMS 2004-F-Marketing - Price/Yield - 2A1A

[illegible]

This Structural Term Sheet, Collateral Term Sheet, or Computational Materials, as appropriate (the "material"), is for your private information and Banc of America Securities LLC (the "Underwriter") is not soliciting any action based upon it. This material is not to be construed as an offer to sell or the solicitation of any offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. This material is based on information that the Underwriter considers reliable, but the Underwriter does not represent that it is accurate or complete and it should not be relied upon as such. By accepting this material the recipient agrees that it will not distribute or provide the material to any other person. The information contained in this material may pertain to securities that ultimately are not sold. The information contained in this material may be based on assumptions regarding market conditions and other matters as reflected herein. The Underwriter makes no representation regarding the reasonableness of such assumptions or the likelihood that any of such assumptions will coincide with actual market conditions or events, and this material should not be relied upon for such purposes. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material may, from time to time, have long or short positions in, and buy and sell, the securities mentioned therein or derivatives thereof (including options). This material may be filed with the Securities and Exchange Commission (the "SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on the material only. Information in this material regarding any securities backing any assets backing any securities discussed herein supersedes all prior information regarding such assets. Any information in the material, whether regarding the assets backing any securities discussed herein or otherwise, will be superseded by the information contained in any final prospectus for any securities actually sold to you. This material is furnished solely by the Underwriter and not by the issuer of the securities. The issuer of the securities has not prepared, reviewed or participated in the preparation of this material, is not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as underwriter and not as agent for the issuer in connection with the proposed transaction.

[illegible]

This Structural Term Sheet, Collateral Term Sheet, or Computational Materials, as appropriate (the "material"), is for your private information and Banc of America Securities LLC (the "Underwriter") is not soliciting any action based upon it. This material is not to be construed as an offer to sell or the solicitation of any offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. The Underwriter does not represent that it is accurate or complete and it should not be relied upon as such. By accepting this material the recipient agrees that it will not distribute or provide the material to any other person. The information contained in this material may pertain to securities that ultimately are not sold. The information contained in this material may coincide with actual market conditions or events, and this material should not be relied upon for such purposes. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material may, from time to time, have long or short positions in, and buy and sell, the securities mentioned therein or derivatives thereof (including options). This material may be utilized with the Securities and Exchange Commission (the "SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on this material only. Information in this material regarding any assets backing any securities discussed herein superseded herein or otherwise, will be superseded by the information contained in any final prospectus for any securities actually sold to you. This material is furnished solely by the Underwriter and not by the issuer of the securities. The issuer of the securities has not prepared, reviewed or participated in the preparation of this material, is not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as agent for the issuer in connection with the proposed transaction.

Balance	\$85,998,000.00	Delay	24
Initial Coupon	4.190648217	Dated	6/1/2004
Settle	6/28/2004	First Payment	7/25/2004

The Structural Term Sheet, Collateral Term Sheet, and Confidential Materials, as appropriate, the "material", is for your private information and Bank of America Securities LLC (the "Underwriter") is not soliciting any action based upon it. The material is not to be construed as an offer to sell or the solicitation of any offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. This material is based on information that the Underwriter considers reliable, but the Underwriter does not represent or warrant that the information is accurate or complete and it should not be relied upon as such. By accepting this material the recipient agrees that it will not distribute or provide the material to any other person. No information contained in this material may pertain to securities that ultimately are not sold. The information contained in this material may be based on assumptions regarding market conditions and other matters as reflected herein. The Underwriter makes no representation regarding the reasonableness of such assumptions or the likelihood that any of such assumptions will coincide with actual market events, and this material should not be relied upon for such purposes. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material may, from time to time, have long or short positions in, and buy and sell, the securities mentioned therein or derivatives thereof (including options). This material may be filed with the Securities and Exchange Commission (the "SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on this material only. Information in this material regarding any assets backing any of the securities mentioned herein is not intended to provide actual data to you, but is intended to provide a general overview of the assets backing any of the securities mentioned herein. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material, may find that the information actually sold to you is different from the information contained in this material. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material, are not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as underwriter and not acting as agent for the issuer in connection with the proposed transaction.

BOAMS 2004-F-Final - Price/Yield - 2A4-A

Balance	\$64,817,000.00	Delay	24								
Initial Coupon	4.190648217	Dated	6/1/2004								
Settle	6/28/2004	First Payment	7/25/2004								
				5 CPR To Roll	10 CPR To Roll	15 CPR To Roll	20 CPR To Roll	25 CPR To Roll	30 CPR To Roll	40 CPR To Roll	50 CPR To Roll
				Yield	Yield	Yield	Yield	Yield	Yield	Yield	Yield
				4.705	4.705	4.734	4.897	5.071	5.255	5.665	6.142
				4.698	4.698	4.726	4.886	5.057	5.238	5.640	6.109
				4.691	4.691	4.717	4.874	5.042	5.220	5.616	6.076
				4.683	4.683	4.708	4.863	5.027	5.202	5.591	6.042
				4.676	4.676	4.699	4.851	5.013	5.184	5.566	6.009
				4.669	4.669	4.691	4.840	4.998	5.167	5.541	5.976
				4.661	4.661	4.682	4.828	4.984	5.149	5.516	5.943
				4.654	4.654	4.673	4.817	4.969	5.131	5.492	5.910
				4.647	4.647	4.664	4.805	4.955	5.113	5.467	5.877
				4.639	4.639	4.656	4.794	4.940	5.096	5.442	5.844
				4.632	4.632	4.647	4.782	4.926	5.078	5.417	5.811
				4.625	4.625	4.638	4.770	4.911	5.060	5.392	5.778
				4.617	4.617	4.630	4.759	4.897	5.043	5.368	5.745
				4.610	4.610	4.621	4.747	4.882	5.025	5.343	5.712
				4.603	4.603	4.612	4.736	4.868	5.007	5.318	5.679
				4.596	4.596	4.603	4.724	4.853	4.990	5.294	5.646
				4.588	4.588	4.595	4.713	4.839	4.972	5.269	5.614
				4.908	4.908	4.053	3.000	2.351	1.913	1.353	1.010
				4.338	4.338	3.642	2.753	2.186	1.793	1.281	0.961

This Structural Term Sheet, Collateral Term Sheet, or Computational Materials, as appropriate (the "material"), is for your private information and Banc of America Securities LLC (the "Underwriter") is not soliciting any action based upon it. This material is not to be construed as an offer to sell or the solicitation of any offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. This material is based on information that the Underwriter considers reliable, but the Underwriter does not represent that it is accurate or complete and it should not be relied upon as such. By accepting this material the recipient agrees that it will not distribute or provide the material to any other person. The information contained in this material may pertain to securities that ultimately are not sold. The information contained in this material may be based on assumptions regarding market conditions and other matters as related herein. The Underwriter makes no representation regarding the reasonableness of such assumptions or the likelihood that any of such assumptions will coincide with actual market conditions or events, and this material should not be relied upon for such purposes. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material may, from time to time, have long or short positions in, and buy and sell, the securities mentioned therein or derivatives thereof (including options). This material may be filed with the Securities and Exchange Commission (the "SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on this material only. Information in this material regarding any assets backing any securities discussed herein supersedes all prior information regarding such assets. Any information in the material, whether regarding the assets backing any securities discussed herein or otherwise, will be superseded by the information contained in any final prospectus for any securities actually sold to you. This material is furnished solely by the Underwriter and not by the issuer of the securities. The issuer of the securities has not prepared, reviewed or participated in the preparation of this material, is not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as underwriter and not acting as agent for the issuer in connection with the proposed transaction.

BOAMS 2004-F-Final - Price/Yield - 2A4-B

Balance	\$64,817,000.00	Delay	24																
Initial Coupon	4.190648217	Dated	6/1/2004																
Settle	6/28/2004	First Payment	7/25/2004																
Price	5 CPR To Roll	Yield	10 CPR To Roll	Yield	15 CPR To Roll	Yield	20 CPR To Roll	Yield	25 CPR To Roll	Yield	30 CPR To Roll	Yield	40 CPR To Roll	Yield	50 CPR To Roll	Yield			
	96-20.48	5.001	5.001	5.087	5.364	5.659	5.972	6.669	7.481										
	96-24.48	4.971	4.971	5.051	5.317	5.600	5.899	6.568	7.346										
	96-28.48	4.942	4.942	5.016	5.270	5.540	5.827	6.467	7.211										
	97-00.48	4.912	4.912	4.980	5.223	5.481	5.755	6.366	7.076										
	97-04.48	4.882	4.882	4.945	5.176	5.423	5.684	6.265	6.942										
	97-08.48	4.853	4.853	4.910	5.130	5.364	5.612	6.165	6.808										
	97-12.48	4.823	4.823	4.875	5.083	5.305	5.540	6.065	6.674										
	97-16.48	4.794	4.794	4.839	5.036	5.246	5.469	5.965	6.541										
	97-20.48	4.764	4.764	4.804	4.990	5.188	5.398	5.865	6.408										
	97-24.48	4.735	4.735	4.769	4.944	5.129	5.326	5.765	6.275										
	97-28.48	4.705	4.705	4.734	4.897	5.071	5.255	5.665	6.142										
	98-00.48	4.676	4.676	4.699	4.851	5.013	5.184	5.566	6.009										
	98-04.48	4.647	4.647	4.664	4.805	4.955	5.113	5.467	5.877										
	98-08.48	4.617	4.617	4.630	4.759	4.897	5.043	5.368	5.745										
	98-12.48	4.588	4.588	4.595	4.713	4.839	4.972	5.269	5.614										
	98-16.48	4.559	4.559	4.560	4.667	4.781	4.902	5.170	5.482										
	98-20.48	4.530	4.530	4.525	4.621	4.723	4.831	5.072	5.351										
	98-24.48	4.501	4.501	4.491	4.575	4.665	4.761	4.973	5.220										
	98-28.48	4.472	4.472	4.456	4.530	4.608	4.691	4.875	5.089										
99-00.48	4.443	4.443	4.422	4.484	4.550	4.621	4.777	4.959											
99-04.48	4.414	4.414	4.387	4.438	4.493	4.551	4.679	4.829											
99-08.48	4.385	4.385	4.353	4.393	4.436	4.481	4.582	4.699											
99-12.48	4.356	4.356	4.318	4.347	4.378	4.411	4.484	4.569											
99-16.48	4.327	4.327	4.284	4.302	4.321	4.342	4.387	4.440											
99-20.48	4.299	4.299	4.250	4.257	4.264	4.272	4.290	4.311											
WAL	4.908	4.908	4.053	3.000	2.351	1.913	1.353	1.010											
Mod Durn	4.338	4.338	3.642	2.753	2.186	1.793	1.281	0.961											
Principal Window Begin	05/25/2009	05/25/2009	11/25/2007	12/25/2006	06/25/2006	01/25/2006	08/25/2005	04/25/2005											
Principal Window End	05/25/2009	05/25/2009	03/25/2009	12/25/2007	03/25/2007	09/25/2006	01/25/2006	08/25/2005											
LIBOR_1YR	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2											

This Structural Term Sheet, Collateral Term Sheet, or Computational Materials, as appropriate (the "material"), is for your private information and Bank of America Securities LLC (the "Underwriter") is not soliciting any action based upon it. This material is not to be construed as an offer to sell or the solicitation of any offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. This material is based on information that the Underwriter considers reliable, but the Underwriter does not represent that it is accurate or complete and it should not be relied upon as such. By accepting this material the recipient agrees that it will not distribute or provide the material to any other person. The information contained in this material may pertain to securities that ultimately are not sold. The information contained in this material may be based on assumptions regarding market conditions and other matters as reflected herein. The Underwriter makes no representation regarding the reasonableness of such assumptions or the likelihood that any of such assumptions will coincide with actual market conditions or events, and this material should not be relied upon for such purposes. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of the material may, from time to time, have long or short positions in, and buy and sell, the securities mentioned herein or derivatives thereof (including options). This material may be filed with the Securities and Exchange Commission (the "SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on the material only. Information in this material regarding any securities discussed herein supersede all prior information regarding such securities. Any information in the material, whether regarding the assets backing any securities discussed herein or otherwise, will be superseded by the information contained in any final prospectus for any securities actually sold to you. This material is furnished solely by the Underwriter and not by the issuer of the securities. The issuer of the securities has not prepared, reviewed or participated in the preparation of this material, is not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as underwriter and not acting as agent for the issuer in connection with the proposed transaction.

BOAMS 2004-F-Final - Price/Yield - 2A5-A

Balance \$65,156,000.00 Delay 24
Initial Coupon 4.190648217 Dated 6/1/2004
Settle 6/28/2004 First Payment 7/25/2004

Price	5 CPR To Roll	10 CPR To Roll	15 CPR To Roll	20 CPR To Roll	25 CPR To Roll	30 CPR To Roll	40 CPR To Roll	50 CPR To Roll
	Yield	Yield	Yield	Yield	Yield	Yield	Yield	Yield
96-25.98	4.960	4.960	4.955	5.019	5.223	5.441	5.921	6.486
96-26.98	4.953	4.953	4.947	5.010	5.212	5.428	5.902	6.461
96-27.98	4.945	4.945	4.940	5.002	5.201	5.415	5.884	6.436
96-28.98	4.938	4.938	4.932	4.993	5.191	5.402	5.865	6.412
96-29.98	4.930	4.930	4.925	4.985	5.180	5.388	5.847	6.387
96-30.98	4.923	4.923	4.917	4.976	5.169	5.375	5.828	6.362
96-31.98	4.916	4.916	4.910	4.967	5.158	5.362	5.810	6.337
97-00.98	4.908	4.908	4.903	4.959	5.147	5.349	5.791	6.313
97-01.98	4.901	4.901	4.895	4.950	5.136	5.335	5.773	6.288
97-02.98	4.893	4.893	4.888	4.941	5.125	5.322	5.754	6.263
97-03.98	4.886	4.886	4.880	4.933	5.115	5.309	5.736	6.239
97-04.98	4.879	4.879	4.873	4.924	5.104	5.296	5.717	6.214
97-05.98	4.871	4.871	4.865	4.916	5.093	5.282	5.699	6.189
97-06.98	4.864	4.864	4.858	4.907	5.082	5.269	5.680	6.165
97-07.98	4.856	4.856	4.851	4.898	5.071	5.256	5.662	6.140
97-08.98	4.849	4.849	4.843	4.890	5.060	5.243	5.643	6.115
97-09.98	4.842	4.842	4.836	4.881	5.050	5.230	5.625	6.091
WAL	4.908	4.908	4.901	4.158	3.244	2.628	1.856	1.380
Mod Durm	4.330	4.330	4.323	3.720	2.956	2.424	1.736	1.300
Principal Window Begin	05/25/2009	05/25/2009	03/25/2009	12/25/2007	03/25/2007	09/25/2006	01/25/2006	08/25/2005
Principal Window End	05/25/2009	05/25/2009	05/25/2009	05/25/2009	04/25/2008	07/25/2007	08/25/2006	01/25/2006
LIBOR_1YR	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2

This Structural Term Sheet, Collateral Term Sheet, or Computational Materials, as appropriate (the "Material"), is for your private information and Banc of America Securities LLC (the "Underwriter") is not soliciting any action based upon it. This material is not to be construed as an offer to sell or the solicitation of any offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. This material is based on information that the Underwriter considers reliable, but the Underwriter does not represent that it is accurate or complete and it should not be relied upon as such. By accepting this material the recipient agrees that it will not distribute or provide the material to any other person. The information contained in this material may coincide with actual market material may be based on assumptions regarding market conditions and other matters as reflected herein. The Underwriter makes no representation regarding the reasonableness of such assumptions or the likelihood that any of such assumptions will coincide with actual market conditions or events, and this material should not be relied upon for such purposes. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material may, from time to time, have long or short positions in, and buy and sell, the securities mentioned therein or derivatives thereof (including options). This material may be filed with the Securities and Exchange Commission (the "SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on this material only. Information in this material regarding any assets backing any securities discussed herein supersedes all prior information regarding such assets. Any information in the material, whether regarding the assets backing any securities discussed herein or otherwise, will be superseded by the information contained in any final prospectus for any securities actually sold to you. This material is furnished solely by the Underwriter and not by the issuer of the securities. The issuer of the securities has not prepared, reviewed or participated in the preparation of this material, is not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as underwriter and not acting as agent for the issuer in connection with the proposed transaction.

BOAMS 2004-F-Final - Price/Yield - 2A5-B

Balance \$65,156,000.00 Delay 24
 Initial Coupon 4.190648217 Dated 6/1/2004
 Settle 6/28/2004 First Payment 7/25/2004

Price	5 CPR To Roll	10 CPR To Roll	15 CPR To Roll	20 CPR To Roll	25 CPR To Roll	30 CPR To Roll	40 CPR To Roll	50 CPR To Roll
	Yield	Yield	Yield	Yield	Yield	Yield	Yield	Yield
95-17.98	5.260	5.260	5.255	5.368	5.662	5.977	6.669	7.486
95-21.98	5.230	5.230	5.225	5.333	5.618	5.923	6.594	7.385
95-25.98	5.199	5.199	5.194	5.298	5.574	5.869	6.518	7.284
95-29.98	5.169	5.169	5.164	5.263	5.530	5.815	6.443	7.184
96-01.98	5.139	5.139	5.134	5.228	5.486	5.762	6.368	7.083
96-05.98	5.109	5.109	5.104	5.193	5.442	5.708	6.293	6.983
96-09.98	5.079	5.079	5.074	5.158	5.398	5.655	6.219	6.884
96-13.98	5.050	5.050	5.044	5.123	5.354	5.601	6.144	6.784
96-17.98	5.020	5.020	5.014	5.088	5.311	5.548	6.069	6.684
96-21.98	4.990	4.990	4.984	5.054	5.267	5.495	5.995	6.585
96-25.98	4.960	4.960	4.955	5.019	5.223	5.441	5.921	6.486
96-29.98	4.930	4.930	4.925	4.985	5.180	5.388	5.847	6.387
97-01.98	4.901	4.901	4.895	4.950	5.136	5.335	5.773	6.288
97-05.98	4.871	4.871	4.865	4.916	5.093	5.282	5.699	6.189
97-09.98	4.842	4.842	4.836	4.881	5.050	5.230	5.625	6.091
97-13.98	4.812	4.812	4.806	4.847	5.006	5.177	5.551	5.993
97-17.98	4.782	4.782	4.777	4.812	4.963	5.124	5.478	5.894
97-21.98	4.753	4.753	4.747	4.778	4.920	5.072	5.404	5.797
97-25.98	4.724	4.724	4.718	4.744	4.877	5.019	5.331	5.699
97-29.98	4.694	4.694	4.688	4.710	4.834	4.967	5.258	5.601
98-01.98	4.665	4.665	4.659	4.676	4.791	4.914	5.185	5.504
98-05.98	4.636	4.636	4.630	4.642	4.748	4.862	5.112	5.407
98-09.98	4.606	4.606	4.600	4.608	4.705	4.810	5.039	5.310
98-13.98	4.577	4.577	4.571	4.574	4.663	4.758	4.967	5.213
98-17.98	4.548	4.548	4.542	4.540	4.620	4.706	4.894	5.116
WAL	4.908	4.908	4.901	4.158	3.244	2.628	1.856	1.380
Mod Durn	4.330	4.330	4.323	3.720	2.956	2.424	1.736	1.300
Principal Window Begin	05/25/2009	05/25/2009	03/25/2009	12/25/2007	03/25/2007	09/25/2006	01/25/2006	08/25/2005
Principal Window End	05/25/2009	05/25/2009	05/25/2009	05/25/2009	04/25/2008	07/25/2007	08/25/2006	01/25/2006
LIBOR_1YR	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2

This Structural Term Sheet, Collateral Term Sheet, or Computational Materials, as appropriate (the "material"), is for your private information and does not constitute an offer to sell or the solicitation of an offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. This material is based on information that the Underwriter deems reliable, but the Underwriter does not warrant that it is accurate or complete and it should not be relied upon as such. By accepting this material, the recipient agrees that it will not distribute or provide this material to any other person. The information contained in this material is for informational purposes only and should not be used as a basis for investment decisions. The Underwriter makes no representation regarding the reasonableness of such assumptions or the accuracy of the information contained herein. The Underwriter and its affiliates, officers, directors, partners and employees, including persons who are associated with the Underwriter, may have long or short positions in, and may buy and sell, the securities mentioned herein or derivatives thereof (including options). This material may be filed with the Securities and Exchange Commission ("SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on this material only. Information in this material regarding any assets backing any securities discussed herein supersedes all prior information regarding such assets. Any information in the material, whether regarding the assets backing any securities discussed herein or otherwise, will be superseded by the information contained in any final prospectus for any securities actually sold to you. This material is furnished solely by the Underwriter and not by the issuer of the securities. The issuer of the securities has not prepared, reviewed or participated in the preparation of this material, is not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as underwriter and not acting as agent for the issuer in connection with the proposed transaction.

BOAMS 2004-F-Final - Price/Yield - 2A6

Balance	\$177,091,000.00	Delay	24
Initial Coupon	4.190648217	Dated	6/1/2004
Settle	6/28/2004	First Payment	7/25/2004

Price	5 CPR To Roll	10 CPR To Roll	15 CPR To Roll	20 CPR To Roll	25 CPR To Roll	30 CPR To Roll	40 CPR To Roll	50 CPR To Roll
	Yield	Yield	Yield	Yield	Yield	Yield	Yield	Yield
95-00.00	5.396	5.396	5.396	5.395	5.417	5.490	5.734	6.097
95-02.00	5.381	5.381	5.381	5.380	5.402	5.473	5.714	6.071
95-04.00	5.366	5.366	5.366	5.365	5.386	5.456	5.694	6.046
95-06.00	5.350	5.350	5.350	5.349	5.370	5.440	5.673	6.020
95-08.00	5.335	5.335	5.335	5.334	5.355	5.423	5.653	5.995
95-10.00	5.320	5.320	5.320	5.319	5.339	5.406	5.633	5.970
95-12.00	5.305	5.305	5.305	5.304	5.323	5.389	5.613	5.944
95-14.00	5.290	5.290	5.290	5.289	5.308	5.373	5.592	5.919
95-16.00	5.275	5.275	5.275	5.274	5.292	5.356	5.572	5.894
95-18.00	5.260	5.260	5.260	5.259	5.277	5.339	5.552	5.868
95-20.00	5.245	5.245	5.245	5.244	5.261	5.322	5.532	5.843
95-22.00	5.229	5.229	5.229	5.229	5.245	5.306	5.512	5.818
95-24.00	5.214	5.214	5.214	5.213	5.230	5.289	5.491	5.793
95-26.00	5.199	5.199	5.199	5.198	5.214	5.272	5.471	5.768
95-28.00	5.184	5.184	5.184	5.183	5.199	5.256	5.451	5.742
95-30.00	5.169	5.169	5.169	5.168	5.183	5.239	5.431	5.717
96-00.00	5.154	5.154	5.154	5.153	5.168	5.223	5.411	5.692
96-02.00	5.139	5.139	5.139	5.138	5.152	5.206	5.391	5.667
96-04.00	5.124	5.124	5.124	5.123	5.137	5.189	5.371	5.642
96-06.00	5.109	5.109	5.109	5.108	5.121	5.173	5.351	5.617
96-08.00	5.094	5.094	5.094	5.093	5.106	5.156	5.331	5.592
96-10.00	5.079	5.079	5.079	5.078	5.091	5.140	5.311	5.567
96-12.00	5.064	5.064	5.064	5.063	5.075	5.123	5.291	5.542
96-14.00	5.049	5.049	5.049	5.048	5.060	5.107	5.271	5.517
96-16.00	5.034	5.034	5.034	5.034	5.044	5.090	5.251	5.492
96-18.00	5.020	5.020	5.020	5.019	5.029	5.074	5.231	5.467
96-20.00	5.005	5.005	5.005	5.004	5.014	5.057	5.211	5.442
96-22.00	4.990	4.990	4.990	4.989	4.998	5.041	5.191	5.417
96-24.00	4.975	4.975	4.975	4.974	4.983	5.024	5.171	5.392
96-26.00	4.960	4.960	4.960	4.959	4.967	5.008	5.151	5.368
96-28.00	4.945	4.945	4.945	4.944	4.952	4.991	5.131	5.343
96-30.00	4.930	4.930	4.930	4.929	4.937	4.975	5.111	5.318
97-00.00	4.915	4.915	4.915	4.914	4.922	4.959	5.092	5.293
97-02.00	4.901	4.901	4.901	4.900	4.906	4.942	5.072	5.268
97-04.00	4.886	4.886	4.886	4.885	4.891	4.926	5.052	5.244
97-06.00	4.871	4.871	4.871	4.870	4.876	4.910	5.032	5.219
97-08.00	4.856	4.856	4.856	4.855	4.860	4.893	5.012	5.194
97-10.00	4.841	4.841	4.841	4.840	4.845	4.877	4.993	5.170
97-12.00	4.827	4.827	4.827	4.826	4.830	4.861	4.973	5.145
97-14.00	4.812	4.812	4.812	4.811	4.815	4.844	4.953	5.120
97-16.00	4.797	4.797	4.797	4.796	4.799	4.828	4.934	5.096
97-18.00	4.782	4.782	4.782	4.781	4.784	4.812	4.914	5.071
97-20.00	4.768	4.768	4.768	4.767	4.769	4.795	4.894	5.047
97-22.00	4.753	4.753	4.753	4.752	4.754	4.779	4.875	5.022
97-24.00	4.738	4.738	4.738	4.737	4.739	4.763	4.855	4.998
97-26.00	4.724	4.724	4.724	4.723	4.724	4.747	4.835	4.973
97-28.00	4.709	4.709	4.709	4.708	4.708	4.730	4.816	4.949
97-30.00	4.694	4.694	4.694	4.693	4.693	4.714	4.796	4.924
98-00.00	4.680	4.680	4.680	4.679	4.678	4.698	4.777	4.900
98-02.00	4.665	4.665	4.665	4.664	4.663	4.682	4.757	4.875
98-04.00	4.650	4.650	4.650	4.649	4.648	4.666	4.737	4.851
98-06.00	4.636	4.636	4.636	4.635	4.633	4.649	4.718	4.827
98-08.00	4.621	4.621	4.621	4.620	4.618	4.633	4.698	4.802
98-10.00	4.606	4.606	4.606	4.605	4.603	4.617	4.679	4.778
98-12.00	4.592	4.592	4.592	4.591	4.588	4.601	4.659	4.754
98-14.00	4.577	4.577	4.577	4.576	4.573	4.585	4.640	4.729
98-16.00	4.563	4.563	4.563	4.562	4.558	4.569	4.621	4.705
98-18.00	4.548	4.548	4.548	4.547	4.543	4.553	4.601	4.681
98-20.00	4.533	4.533	4.533	4.532	4.528	4.537	4.582	4.657
98-22.00	4.519	4.519	4.519	4.518	4.513	4.521	4.562	4.633
98-24.00	4.504	4.504	4.504	4.503	4.498	4.505	4.543	4.608
98-26.00	4.490	4.490	4.490	4.489	4.483	4.489	4.524	4.584
98-28.00	4.475	4.475	4.475	4.474	4.468	4.472	4.504	4.560
98-30.00	4.461	4.461	4.461	4.460	4.453	4.456	4.485	4.536
99-00.00	4.446	4.446	4.446	4.445	4.438	4.440	4.466	4.512
WAL	4.908	4.908	4.908	4.908	4.745	4.407	3.604	2.855
Mod Dum	4.329	4.329	4.329	4.329	4.197	3.917	3.240	2.594
Principal Window Begin	5/25/2009	5/25/2009	5/25/2009	5/25/2009	4/25/2008	7/25/2007	8/25/2006	1/25/2006
Principal Window End	5/25/2009	5/25/2009	5/25/2009	5/25/2009	5/25/2009	5/25/2009	5/25/2009	5/25/2009

LIBOR_1YR	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
-----------	-----	-----	-----	-----	-----	-----	-----	-----

This Structural Term Sheet, Collateral Term Sheet, or Computational Materials, as appropriate (the "material"), is for your private information and Banc of America Securities LLC (the "Underwriter") is not soliciting any action based upon it. This material is not to be construed as an offer to sell or the solicitation of any offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. This material is based on information that the Underwriter considers reliable, but the Underwriter does not represent that it is accurate or complete and it should not be relied upon as such. By accepting this material the recipient agrees that it will not distribute or provide the material to any other person. The information contained in this material may pertain to securities that ultimately are not sold. The information contained in this material may be based on assumptions regarding market conditions and other matters as reflected herein. The Underwriter makes no representation regarding the reasonableness of such assumptions or the likelihood that any of such assumptions will coincide with actual market conditions or events, and this material should not be relied upon for such purposes. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material may, from time to time, have long or short positions in, and buy and sell, the securities mentioned therein or derivatives thereof (including options). This material may be filed with the Securities and Exchange Commission (the "SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on this material only. Information in this material regarding any assets backing any securities discussed herein supersedes all prior information regarding such assets. Any information in the materials whether regarding the assets backing any securities discussed herein or otherwise, will be superseded by the information contained in any final prospectus for any securities actually sold to you. This material is furnished solely by the Underwriter and not by the issuer of the securities. The issuer of the securities has not prepared, reviewed or participated in the preparation of this material, is not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as underwriter and not acting as agent for the issuer in connection with the proposed transaction.

BOAMS 2004-F-Final - Price/Yield - 2A6-A

Balance \$177,091,000.00 Delay 24
 Initial Coupon 4.190648217 Dated 6/1/2004
 Settle 6/28/2004 First Payment 7/25/2004

Price	5 CPR To Roll	10 CPR To Roll	15 CPR To Roll	20 CPR To Roll	25 CPR To Roll	30 CPR To Roll	40 CPR To Roll	50 CPR To Roll
	Yield	Yield	Yield	Yield	Yield	Yield	Yield	Yield
96-30.41	4.927	4.927	4.927	4.926	4.934	4.972	5.107	5.313
96-31.41	4.920	4.920	4.920	4.919	4.926	4.964	5.098	5.301
97-00.41	4.912	4.912	4.912	4.911	4.918	4.955	5.088	5.288
97-01.41	4.905	4.905	4.905	4.904	4.911	4.947	5.078	5.276
97-02.41	4.898	4.898	4.898	4.897	4.903	4.939	5.068	5.263
97-03.41	4.890	4.890	4.890	4.889	4.895	4.931	5.058	5.251
97-04.41	4.883	4.883	4.883	4.882	4.888	4.923	5.048	5.239
97-05.41	4.875	4.875	4.875	4.874	4.880	4.914	5.038	5.226
97-06.41	4.868	4.868	4.868	4.867	4.873	4.906	5.028	5.214
97-07.41	4.861	4.861	4.861	4.860	4.865	4.898	5.018	5.202
97-08.41	4.853	4.853	4.853	4.852	4.857	4.890	5.008	5.189
97-09.41	4.846	4.846	4.846	4.845	4.850	4.882	4.999	5.177
97-10.41	4.838	4.838	4.838	4.837	4.842	4.874	4.989	5.165
97-11.41	4.831	4.831	4.831	4.830	4.834	4.865	4.979	5.152
97-12.41	4.824	4.824	4.824	4.823	4.827	4.857	4.969	5.140
97-13.41	4.816	4.816	4.816	4.815	4.819	4.849	4.959	5.128
97-14.41	4.809	4.809	4.809	4.808	4.812	4.841	4.949	5.115
WAL	4.908	4.908	4.908	4.908	4.745	4.407	3.604	2.855
Mod Durm	4.331	4.331	4.331	4.331	4.198	3.919	3.242	2.596
Principal Window Begin	05/25/2009	05/25/2009	05/25/2009	05/25/2009	04/25/2008	07/25/2007	08/25/2006	01/25/2006
Principal Window End	05/25/2009	05/25/2009	05/25/2009	05/25/2009	05/25/2009	05/25/2009	05/25/2009	05/25/2009
LIBOR_1YR	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2

This Structural Term Sheet, Collateral Term Sheet, or Computational Materials, as appropriate (the "material"), is for your private information and Banc of America Securities LLC (the "Underwriter") is not soliciting any action based upon it. This material is not to be construed as an offer to sell or the solicitation of any offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. This material is based on information that the Underwriter considers reliable, but the Underwriter does not represent that it is accurate or complete and it should not be relied upon as such. By accepting this material the recipient agrees that it will not distribute or provide the material to any other person. The information contained in this material may pertain to securities that ultimately are not sold. The information contained in this material may be based on assumptions regarding market conditions and other matters as reflected herein. The Underwriter makes no representation regarding the reasonableness of such assumptions or the likelihood that any of such assumptions will coincide with actual market conditions or events, and this material should not be relied upon for such purposes. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material may, from time to time, have long or short positions in, and buy and sell, the securities mentioned therein or derivatives thereof (including options). This material may be filed with the Securities and Exchange Commission (the "SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on this material only. Information regarding any assets backing any securities discussed herein supersedes all prior information regarding such assets. Any information in the material, whether regarding the assets backing any securities discussed herein or otherwise, will be superseded by the information contained in any final prospectus for any securities actually sold to you. This material is furnished solely by the Underwriter and not by the issuer of the securities. The issuer of the securities has not prepared, reviewed or participated in the preparation of this material, is not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as underwriter and not acting as agent for the issuer in connection with the proposed transaction.

BOAMS 2004-F-Final - Price/Yield - 4A1

Balance \$105,839,000.00 Delay 24
Initial Coupon 4.657028114 Dated 6/1/2004
Settle 6/28/2004 First Payment 7/25/2004

Price	5 CPR To Roll	10 CPR To Roll	15 CPR To Roll	20 CPR To Roll	25 CPR To Roll	30 CPR To Roll	40 CPR To Roll	50 CPR To Roll
	Yield	Yield	Yield	Yield	Yield	Yield	Yield	Yield
98-00.00	5.070	5.134	5.208	5.291	5.385	5.491	5.743	6.055
98-01.00	5.063	5.126	5.198	5.280	5.372	5.476	5.723	6.029
98-02.00	5.056	5.118	5.189	5.269	5.359	5.461	5.704	6.004
98-03.00	5.049	5.110	5.179	5.257	5.346	5.447	5.684	5.979
98-04.00	5.042	5.102	5.170	5.246	5.333	5.432	5.664	5.953
98-05.00	5.036	5.094	5.161	5.235	5.321	5.417	5.645	5.928
98-06.00	5.029	5.086	5.151	5.224	5.308	5.402	5.625	5.902
98-07.00	5.022	5.078	5.142	5.213	5.295	5.387	5.606	5.877
98-08.00	5.015	5.070	5.132	5.202	5.282	5.373	5.586	5.851
98-09.00	5.008	5.062	5.123	5.191	5.270	5.358	5.567	5.826
98-10.00	5.001	5.054	5.113	5.180	5.257	5.343	5.547	5.801
98-11.00	4.994	5.046	5.104	5.169	5.244	5.328	5.528	5.775
98-12.00	4.988	5.038	5.094	5.159	5.231	5.314	5.508	5.750
98-13.00	4.981	5.030	5.085	5.148	5.219	5.299	5.489	5.725
98-14.00	4.974	5.021	5.076	5.137	5.206	5.284	5.470	5.700
98-15.00	4.967	5.013	5.066	5.126	5.193	5.270	5.450	5.674
98-16.00	4.960	5.005	5.057	5.115	5.180	5.255	5.431	5.649
WAL	5.483	4.623	3.916	3.336	2.851	2.445	1.823	1.381
Mod Durm	4.626	3.932	3.358	2.883	2.484	2.148	1.626	1.250
Principal Window Begin	07/25/2004	07/25/2004	07/25/2004	07/25/2004	07/25/2004	07/25/2004	07/25/2004	07/25/2004
Principal Window End	05/25/2011	05/25/2011	05/25/2011	05/25/2011	05/25/2011	05/25/2011	05/25/2011	05/25/2011
LIBOR_1YR	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2

This Structural Term Sheet, Collateral Term Sheet, or Computational Materials, as appropriate (the "material"), is for your private information and Banc of America Securities LLC (the "Underwriter") is not soliciting any action based upon it. This material is not to be construed as an offer to sell or the solicitation of any offer to buy any security in any jurisdiction where such an offer or solicitation would be illegal. This material is based on information that the Underwriter considers reliable, but the Underwriter does not represent that it is accurate or complete and it should not be relied upon as such. By accepting this material the recipient agrees that it will not distribute or provide the material to any other person. The information contained in this material may pertain to securities that ultimately are not sold. The information contained in this material may be based on assumptions regarding market conditions and other matters as reflected herein. The Underwriter makes no representation regarding the reasonableness of such assumptions or the likelihood that any of such assumptions will coincide with actual market conditions or events, and this material should not be relied upon for such purposes. The Underwriter and its affiliates, officers, directors, partners and employees, including persons involved in the preparation or issuance of this material may, from time to time, have long or short positions in, and buy and sell, the securities mentioned therein or derivatives thereof (including options). This material may be filed with the Securities and Exchange Commission (the "SEC") and incorporated by reference into an effective registration statement previously filed with the SEC under Rule 415 of the Securities Act of 1933, including all cases where the material does not pertain to securities that are ultimately offered for sale pursuant to such registration statement. Information contained in this material is current as of the date appearing on this material only. Information in this material regarding any securities discussed herein supersedes all prior information regarding such assets. Any information in the material, whether regarding the assets backing any securities discussed herein or otherwise, will be superseded by any final prospectus for any securities actually sold to you. This material is furnished solely by the Underwriter and not by the issuer of the securities. The issuer of the securities has not prepared, reviewed or participated in the preparation of this material, is not responsible for the accuracy of this material and has not authorized the dissemination of this material. The Underwriter is acting as underwriter and not acting as agent for the issuer in connection with the proposed transaction.

b04f_roll.cdi
! B04F_ROLL.CDI #CMOVER_3.0E WHOLE_LOAN ! MAX_CF_VECTSIZE 620

!! Created by Intex Deal Maker v3.7.011 , subroutines 3.1
!! 06/06/2004 4:36 PM

! Modeled in the Intex CMO Modeling Language, (B000BDB448D23)
! 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.

!The tables and other statistical analyses (the "Hypothetical Performance Data")
that you will produce using Intex with the attached information are privileged and
intended solely for use by you

!(the party to whom Banc of America Securities LLC provided the computer file source
code used to generate them). The Hypothetical Performance Data will be generated by
you using a computer file source

!code prepared by Banc of America Securities LLC in reliance upon information
furnished by the issuer of the securities and its affiliates, the accuracy and
completeness of which has not been verified by

!Banc of America Securities LLC or any other person. The computer file source code
that you will use to prepare the Hypothetical Performance Data was furnished to you
solely by Banc of America Securities LLC

!and not by the issuer of the securities. It may not be (a) used for any purpose
other than to make a preliminary evaluation of the referenced securities or (b)
provided by you to any third party other than your

!legal, tax, financial and/or accounting advisors for the purposes of evaluating the
Hypothetical Performance Data. You agree that the Hypothetical Performance Data
will be generated by or on behalf of you,

!and that neither Banc of America Securities LLC nor anyone acting on its behalf has
generated or is in any way responsible for any Hypothetical Performance Data.

!Numerous assumptions were used in preparing the computer file source code you will
use to generate the Hypothetical Performance Data. Those assumptions may or may not
be reflected in the Hypothetical

!Performance Data. As such, no assurance can be given as to the Hypothetical
Performance Data's accuracy, appropriateness or completeness in any particular
context; nor as to whether the Hypothetical

!Performance Data and/or the assumptions upon which it is based reflect present
market conditions or future market performance. The Hypothetical Performance Data
should not be construed as either projections

!or predictions or as legal, tax, financial or accounting advice.

!Any weighted average lives, yields and principal payment periods shown in the
Hypothetical Performance Data will be based on prepayment assumptions, and changes
in such prepayment assumptions may

!dramatically affect such weighted average lives, yields and principal payment
periods. In addition, it is possible that prepayments on the underlying assets will
occur at rates slower or faster than the rates

!shown in the Hypothetical Performance Data. Furthermore, unless otherwise
provided, the Hypothetical Performance Data assumes no losses on the underlying
assets and no interest shortfall. The specific

!characteristics of the securities may differ from those shown in the Hypothetical
Performance Data due to, among other things, differences between (a) the actual
underlying assets and the hypothetical

!underlying assets used in preparing the Hypothetical Performance Data and (b) the
assumptions used by you in producing the Hypothetical Performance Data and the
actual assumptions used in pricing the

!actual securities. The principal amount, designation and terms of any security
described in the Hypothetical Performance Data are subject to change prior to
issuance. You should contact the Banc of America

!Securities LLC Trading Desk at (704) 388-1579 to confirm the final principal
amount, designation and terms of any security described in this communication prior
to committing to purchase that security. Neither

b04f_roll.cdi

!Banc of America Securities LLC nor any of its affiliates makes any representation or warranty as to the actual rate or timing of payments on any of the underlying assets or the payments or yield on the securities.

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

!Prospective purchasers are referred to the final prospectus supplement relating to the securities discussed in this communication for definitive yield and maturity information regarding those securities, based
!on the final principal amounts, designations and terms of those securities. Once available, a final prospectus and prospectus supplement may be obtained by contacting the Banc of America Securities LLC
!Trading Desk at (704) 388-1579.

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

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

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

COLLAT_GROUPS 1 2 3 4

!
!

```
DEFINE CONSTANT #OrigCollBal = 1644988940.62
DEFINE CONSTANT #OrigCollBal1 = 229366875.86
DEFINE CONSTANT #OrigCollBal2 = 1091283882.96
DEFINE CONSTANT #OrigCollBal3 = 215281347.61
DEFINE CONSTANT #OrigCollBal4 = 109056834.19
```

!

```
DEFINE CONSTANT #OrigBondBal = 1644988940.62
DEFINE CONSTANT #OrigBondBal1 = 229366875.86
DEFINE CONSTANT #OrigBondBal2 = 1091283882.96
DEFINE CONSTANT #OrigBondBal3 = 215281347.61
DEFINE CONSTANT #OrigBondBal4 = 109056834.19
```

!

```
DEFINE #BondBal1 = 271128940.62
DEFINE #BondBal2 = 1107619840.62
DEFINE #BondBal3 = 257458840.62
DEFINE #BondBal4 = 154367840.62
DEFINE #BondBal = 1790575462.48
```

!

FULL_DEALNAME: b04f_roll

!

```

b04f_roll.cdi
DEAL SIZE:                $ 1644988940.62
PRICING SPEED:            20% CPR
ISSUE DATE:               20040601
SETTLEMENT DATE:          20040628

Record date delay:        24

DEFINE TR_INDEXDEPS_ALL

DEFINE SCHEDULE "SHIFT1%", "SHIFT2%", "SHIFT3%", "SHIFT4%"

TRUSTEE_FEE    GROUP 1      0.002
TRUSTEE_FEE    GROUP 2      0.002
TRUSTEE_FEE    GROUP 3      0.002
TRUSTEE_FEE    GROUP 4      0.002

DEAL_CLOCK_INFO _
ISSUE_CDU_DATE      20040601 _
DEAL_FIRSTPAY_DATE  20040725

DEFINE DYNAMIC STICKY #NetRate = ( COLL_I_MISC("COUPON") ) / COLL_PREV_BAL * 1200
DEFINE DYNAMIC STICKY #NetRate1 = ( COLL_I_MISC("COUPON",1) ) / COLL_PREV_BAL(1)
* 1200
DEFINE DYNAMIC STICKY #NetRate2 = ( COLL_I_MISC("COUPON",2) ) / COLL_PREV_BAL(2)
* 1200
DEFINE DYNAMIC STICKY #NetRate3 = ( COLL_I_MISC("COUPON",3) ) / COLL_PREV_BAL(3)
* 1200
DEFINE DYNAMIC STICKY #NetRate4 = ( COLL_I_MISC("COUPON",4) ) / COLL_PREV_BAL(4)
* 1200

DEFINE TABLE "SI_LOSSAagg" (6, 2) = "MONTH" "SHIFTR"
36.1 20%
96.1 30%
108.1 35%
120.1 40%
132.1 45%
144.1 50%

DEFINE TABLE "SI_2TIMESLOSSAgg" (6, 2) = "MONTH" "PCT"
36.1 20%
96.1 30%
108.1 35%
120.1 40%
132.1 45%
144.1 50%

DEFINE TABLE "CROSSPREP_2TIMESAgg" (1, 2) = "MONTH" "MULTIPLE"
360.1 2

DEFINE #COUPON_SUBS = 0

TOLERANCE CLEANUP 0.00
TOLERANCE WRITEDOWN_OLOSS 1.00

```

```

                                b04f_roll.cdi
INITIAL INDEX    LIBOR_1YR      2.2
!
DEFINE TRANCHE "SUBORD_1", "SUBORD_2", "SUBORD_3", "SUBORD_4", "RESID", "1A1",
"2A1", "GMC_2A1", "2A2", "GMC_2A2", "2A3", "GMC_2A3", "2A4", "GMC_2A4", "2A5",
"GMC_2A5", "2A6", "GMC_2A6", "2A7", "3A1", "4A1", "SUBS", "2WAC-IO"
!
!
Tranche "SUBORD_1" MODELING EXCHANGE
  Block 6766775.86 FLOAT GROUP 1 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate1 )
    0      999
!
Tranche "SUBORD_2" MODELING EXCHANGE
  Block 32192882.96 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0      999
!
Tranche "SUBORD_3" MODELING EXCHANGE
  Block 6351347.61 FLOAT GROUP 3 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate3 )
    0      999
!
Tranche "SUBORD_4" MODELING EXCHANGE
  Block 3217834.19 FLOAT GROUP 4 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate4 )
    0      999
!
Tranche "RESID" SEN_WAC
  Block 100.00 FLOAT GROUP 1 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate1 )
    0      999
!
Tranche "1A1" SEN_WAC
  Block 222600000.00 FLOAT GROUP 1 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate1 )
    0      999
!
Tranche "2A1" SEN_FLT_NO
  Block 135544000.00 GROUP 2 FREQ M FLOAT RESET M _
    DAYCOUNT 30360 BUSINESS_DAY NONE _
    Delay 0 Dated 20040625 Next 20040725
    (IF CURDATE LE 20090525 THEN min(2.341,COLL_NETRATE(2)) ELSE COLL_NETRATE(2))
    0      999
!
Tranche "GMC_2A1" EXCHANGE SEN_GMC_FLT
  Block 135544000.00 GROUP 2 FREQ M FLOAT RESET M _
    DAYCOUNT 30360 BUSINESS_DAY NONE _
    Delay 0 Dated 20040625 Next 20040725
    (IF CURDATE LE 20090525 THEN min(2.341,COLL_NETRATE(2)) ELSE COLL_NETRATE(2))
    0      999
!

```

b04f_roll.cdi

```
Tranche "2A2" SEN_FLT_NO
Block 60180000.00 GROUP 2  FREQ M FLOAT RESET M _
    DAYCOUNT 30360 BUSINESS_DAY NONE _
    Delay 0 Dated 20040625 Next 20040725
    (IF CURDATE LE 20090525 then MIN(3.531,COLL_NETRATE(2)) ELSE COLL_NETRATE(2))
    0 999
!
Tranche "GMC_2A2" EXCHANGE SEN_GMC_FLT
Block 60180000.00 GROUP 2  FREQ M FLOAT RESET M _
    DAYCOUNT 30360 BUSINESS_DAY NONE _
    Delay 0 Dated 20040625 Next 20040725
    (IF CURDATE LE 20090525 then MIN(3.531,COLL_NETRATE(2)) ELSE COLL_NETRATE(2))
    0 999
!
Tranche "2A3" SEN_WAC_NO
Block 101379000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0 999
!
Tranche "GMC_2A3" EXCHANGE SEN_GMC_WAC
Block 101379000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0 999
!
Tranche "2A4" SEN_WAC_NO
Block 76411000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0 999
!
Tranche "GMC_2A4" EXCHANGE SEN_GMC_WAC
Block 76411000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0 999
!
Tranche "2A5" SEN_WAC_NO
Block 76789000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0 999
!
Tranche "GMC_2A5" EXCHANGE SEN_GMC_WAC
Block 76789000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0 999
!
Tranche "2A6" SEN_WAC_NO
Block 208788000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0 999
!
```

```

                                b04f_roll.cdi
Tranche "GMC_2A6" EXCHANGE SEN_GMC_WAC
  Block 208788000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0 999
!
Tranche "2A7" SEN_WAC
  Block 400000000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0 999
!
Tranche "3A1" SEN_WAC
  Block 208930000.00 FLOAT GROUP 3 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate3 )
    0 999
!
Tranche "4A1" SEN_WAC
  Block 105839000.00 FLOAT GROUP 4 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate4 )
    0 999
!
Tranche "SUBS" JUN_WAC
  Block 48528840.62 FLOAT _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #COUPON_SUBS )
    0 999
!
Tranche "2WAC-IO" SEN_FLT_IO
  Block 195724000.00 GROUP 2 FREQ M FLOAT RESET M NOTIONAL WITH FORMULA BEGIN ( IF
CURMONTH LE 59 THEN BBAL("GMC_2A1#1","GMC_2A2#1") ELSE 0 ); _
59 THEN BBAL("GMC_2A1#1","GMC_2A2#1") ELSE 0 ); _
    DAYCOUNT 30360 BUSINESS_DAY NONE _
    Delay 24 Dated 20040601 Next 20040725
    (IF CURDATE LE 20090525 THEN
((COLL_NETRATE(2)-0)*BBAL("2A1","2A2"))-COUPON("2A1")*BBAL("2A1") -
COUPON("2A2")*BBAL("2A2"))/BBAL("2A1","2A2") else 0)
    0 999
!
!
DEFINE PSEUDO_TRANCHE COLLAT _
  Delay 24 Dated 20040601 Next 20040725 settle 20040628
DEFINE PSEUDO_TRANCHE COLLAT GROUP 1 _
  Delay 24 Dated 20040601 Next 20040725 settle 20040628
DEFINE PSEUDO_TRANCHE COLLAT GROUP 2 _
  Delay 24 Dated 20040601 Next 20040725 settle 20040628
DEFINE PSEUDO_TRANCHE COLLAT GROUP 3 _
  Delay 24 Dated 20040601 Next 20040725 settle 20040628
DEFINE PSEUDO_TRANCHE COLLAT GROUP 4 _
  Delay 24 Dated 20040601 Next 20040725 settle 20040628
!
CREDIT_SUPPORT_BASIS GROUP_DEAL
!
ifdef #cmover_3.0g2 _
  DEFINE MACRO #BalanceRatio[1] = BALANCE_RATIO {#1}

```

```

!
ifndef #cmove_3.0g2 _
  DEFINE MACRO #BalanceRatio[1] =
  !
    CLASS "RESID"      NO_BUILD_TRANCHE _
                        = "RESID"
    CLASS "1A1"        NO_BUILD_TRANCHE _
                        = "1A1"
    CLASS "2WAC-IO"    NO_BUILD_TRANCHE _
                        = "2WAC-IO"
    CLASS "2A1"        NO_BUILD_TRANCHE _
                        = "2A1"
    CLASS "2A2"        NO_BUILD_TRANCHE _
                        = "2A2"
    CLASS "2A3"        NO_BUILD_TRANCHE _
                        = "2A3"
    CLASS "2A4"        NO_BUILD_TRANCHE _
                        = "2A4"
    CLASS "2A5"        NO_BUILD_TRANCHE _
                        = "2A5"
    CLASS "2A6"        NO_BUILD_TRANCHE _
                        = "2A6"
    CLASS "2A7"        NO_BUILD_TRANCHE _
                        = "2A7"
    CLASS "3A1"        NO_BUILD_TRANCHE _
                        = "3A1"
    CLASS "4A1"        NO_BUILD_TRANCHE _
                        = "4A1"
    CLASS "SUBORD_1"    DISTRIB_CLASS RULES _
                        = "SUBORD_1"
    CLASS "SUBORD_2"    DISTRIB_CLASS RULES _
                        = "SUBORD_2"
    CLASS "SUBORD_3"    DISTRIB_CLASS RULES _
                        = "SUBORD_3"
    CLASS "SUBORD_4"    DISTRIB_CLASS RULES _
                        = "SUBORD_4"
    CLASS "SEQ" DISTRIB_CLASS PRORATA WRITEDOWN_BAL PRORATA _
                        = "2A1" "2A2" "2A3" "2A4" "2A5" "2A6"
    CLASS "PT" _
                        = "2A7"
    CLASS "SUBS"        NO_BUILD_TRANCHE _
                        = "SUBS", _
    COMBINE_CLASSES {#BalanceRatio}{1.0} = "SUBORD_1" "SUBORD_2" "SUBORD_3"
    "SUBORD_4"
    CLASS "SNR_1"        WRITEDOWN_BAL PRORATA ALLOCATION WRITEDOWN_LIMIT BALANCE
    (#OrigCollBal1); _
                        = "RESID" "1A1" , SUPPORT_CLASSES = "SUBS"
    CLASS "SNR_2"        WRITEDOWN_BAL PRORATA ALLOCATION WRITEDOWN_LIMIT BALANCE
    (#OrigCollBal2); _
                        = "SEQ" "PT" "2WAC-IO", SUPPORT_CLASSES = "SUBS"
    CLASS "SNR_3"        WRITEDOWN_BAL PRORATA ALLOCATION WRITEDOWN_LIMIT BALANCE
    (#OrigCollBal3); _
                        = "3A1" , SUPPORT_CLASSES = "SUBS"
    CLASS "SNR_4"        WRITEDOWN_BAL PRORATA ALLOCATION WRITEDOWN_LIMIT BALANCE
    (#OrigCollBal4); _
                        = "4A1" , SUPPORT_CLASSES = "SUBS"
  !
  !
  CLASS "GRP1" _
                        DISTRIB_CLASS RULES _
                        = "SNR_1" "SUBORD_1"
  CLASS "GRP2" _
                        DISTRIB_CLASS RULES _

```

```

                                b04f_roll.cdi
                                = "SNR_2" "SUBORD_2"
CLASS "GRP3" -
    DISTRIB_CLASS RULES
    = "SNR_3" "SUBORD_3"
CLASS "GRP4" -
    DISTRIB_CLASS RULES
    = "SNR_4" "SUBORD_4"
!
CLASS "ROOT" ROOT_LIST = "GRP1" "GRP2" "GRP3" "GRP4"
!
GROUP 0      ROOT      = 1 2 3 4
!

CLASS "GMC_2A1"          PSEUDO NO_BUILD_TRANCHE = "GMC_2A1#1"
CLASS "GMC_2A2"          PSEUDO NO_BUILD_TRANCHE = "GMC_2A2#1"
CLASS "GMC_2A3"          PSEUDO NO_BUILD_TRANCHE = "GMC_2A3#1"
CLASS "GMC_2A4"          PSEUDO NO_BUILD_TRANCHE = "GMC_2A4#1"
CLASS "GMC_2A5"          PSEUDO NO_BUILD_TRANCHE = "GMC_2A5#1"
CLASS "GMC_2A6"          PSEUDO NO_BUILD_TRANCHE = "GMC_2A6#1"
CLASS "GMC_2A7"          PSEUDO NO_BUILD_TRANCHE = "2A7"
!
CLASS "GMC_SEQ" PSEUDO = "GMC_2A1" "GMC_2A2" "GMC_2A3" "GMC_2A4" "GMC_2A5"
"GMC_2A6"
CLASS "GMC_PT" PSEUDO = "GMC_2A7"
CLASS "GMC_SNR_2" PSEUDO = "GMC_SEQ" "PT" "2WAC-IO"
!
!
DEFINE PSEUDO_TRANCHE CLASS "SNR_1"          Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SUBORD_1"        Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SNR_2"          Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SUBORD_2"        Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SNR_3"          Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SUBORD_3"        Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SNR_4"          Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SUBORD_4"        Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SEQ"            Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "PT"            Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "GMC_SEQ"        Delay 24 Dated 20040601 Next
20040725 DAYCOUNT 30360 BUSINESS_DAY NONE
DEFINE PSEUDO_TRANCHE CLASS "GMC_PT"        Delay 24 Dated 20040601 Next
20040725 DAYCOUNT 30360 BUSINESS_DAY NONE
DEFINE PSEUDO_TRANCHE CLASS "GMC_SNR_2"    Delay 24 Dated 20040601 Next
20040725 DAYCOUNT 30360 BUSINESS_DAY NONE

```

```

!
CROSSOVER when 0
!
DEFINE DYNAMIC STICKY #OrigSubBal1 = #OrigCollBal1 - ORIG_BBAL("SNR_1")
DEFINE DYNAMIC STICKY #SubBal1 = MAX(0, COLL_PREV_BAL(1) - BBAL("SNR_1"))
DEFINE DYNAMIC STICKY #OrigSubBal2 = #OrigCollBal2 - ORIG_BBAL("SNR_2")
DEFINE DYNAMIC STICKY #SubBal2 = MAX(0, COLL_PREV_BAL(2) - BBAL("SNR_2"))
DEFINE DYNAMIC STICKY #OrigSubBal3 = #OrigCollBal3 - ORIG_BBAL("SNR_3")
DEFINE DYNAMIC STICKY #SubBal3 = MAX(0, COLL_PREV_BAL(3) - BBAL("SNR_3"))
DEFINE DYNAMIC STICKY #OrigSubBal4 = #OrigCollBal4 - ORIG_BBAL("SNR_4")
DEFINE DYNAMIC STICKY #SubBal4 = MAX(0, COLL_PREV_BAL(4) - BBAL("SNR_4"))
DEFINE DYNAMIC STICKY #ReduceTestAagg = LOOKUP_TBL( "STEP", CURMONTH ,
"SI_LOSSAagg", "MONTH", "SHIFTR" )
DEFINE DYNAMIC STICKY #Sub2TimesLossPctAgg = LOOKUP_TBL( "STEP", CURMONTH ,
"SI_2TIMESLOSSAagg", "MONTH", "PCT" )
!
TRIGGER "CumLossAgg" _
    ORIG_TESTVAL 0.00% _
    TESTVAL (DELINQ_LOSS_ACCUM/ ORIG_BBAL("SUBS")); _
    TARGETVAL ( #ReduceTestAagg ); _
    ORIG_TARGETVAL NO_CHECK 20% _
    EFFECTIVE_WHEN ( IF CURMONTH GT 84 THEN TRIG_EFFECTIVE_YES ELSE
TRIG_EFFECTIVE_ALWAYSPASS ); _
    TRIGVAL LODIFF
!
TRIGGER "ShiftTriggerAgg" _
    FULL_NAME "Shifting Interest Loss Trigger" _
    DEFINITION "A shifting Interest Loss Trigger will _
exist after month 84 if ;_
        cumulative losses as a percentage of the original subordinate _
bond balance are greater than the percentage in the following table; _
            Month <= %; _
                36 20%; _
                96 30%; _
                108 35%; _
                120 40%; _
                132 45%; _
                144 50% _
The above calculations will be done based on all crossed mortgage loan groups." _
    EFFECTIVE_WHEN ( IF CURMONTH GT 84 THEN TRIG_EFFECTIVE_YES ELSE
TRIG_EFFECTIVE_ALWAYSPASS ); _
    IMPACT "Unscheduled principal that was intended to be paid to the _
subordinate bonds will now be directed to the senior bonds." _
    TRIGVAL FORMULA (TRIGGER("CumLossAgg","TRIGVAL"));
!
!
TRIGGER "TwoTimesCumLossAgg" _
    ORIG_TESTVAL 0.00% _
    TESTVAL (DELINQ_LOSS_ACCUM/(#OrigSubBal1 + #OrigSubBal2 +
#OrigSubBal3 + #OrigSubBal4)); _
    TARGETVAL ( #Sub2TimesLossPctAgg ); _
    ORIG_TARGETVAL NO_CHECK 20% _
    TRIGVAL LODIFF
!
TRIGGER "TwoTimesTriggerAgg" _
    FULL_NAME "Sub Two Times Loss Trigger" _
    DEFINITION "A Sub Two Times Loss Trigger will _
exist if ;_
        cumulative losses as a percentage of the original subordinate _
bond balance are greater than the percentage in the following table; _
            Month <= %; _
                36 20%; _
                96 30%; _

```

```

                                b04f_roll.cdi
                                108      35%;  -
                                120      40%;  -
                                132      45%;  -
                                144      50%;  -
IMPACT      "Unscheduled principal that was intended to be paid to the _
subordinate bonds will be directed to the senior bonds earlier than normal." _
TRIGVAL FORMULA (TRIGGER("TwoTimesCumLossAgg","TRIGVAL"));
!
!
DEFINE DYNAMIC STICKY #COUPON_SUBS = OPTIMAL_INTPMT("SUBORD_1", "SUBORD_2",
"SUBORD_3", "SUBORD_4") / BBAL("SUBORD_1", "SUBORD_2", "SUBORD_3", "SUBORD_4") *
36000 / NDAYS_ACCRUE_INT("SUBS#1")
!
OPTIONAL REDEMPTION:      "Grp_1_Call" _
                           DATE 20070501 _
                           TARGET GROUP 1 _
                           PRICE_P ( COLL_BAL(1) ); _
                           DISTR_P RULES "OPTR_GROUP_1"
!
OPTIONAL REDEMPTION:      "Grp_2_Call" _
                           DATE 20090501 _
                           TARGET GROUP 2 _
                           PRICE_P ( COLL_BAL(2) ); _
                           DISTR_P RULES "OPTR_GROUP_2"
!
OPTIONAL REDEMPTION:      "Grp_3_Call" _
                           DATE 20090501 _
                           TARGET GROUP 3 _
                           PRICE_P ( COLL_BAL(3) ); _
                           DISTR_P RULES "OPTR_GROUP_3"
!
OPTIONAL REDEMPTION:      "Grp_4_Call" _
                           DATE 20110501 _
                           TARGET GROUP 4 _
                           PRICE_P ( COLL_BAL(4) ); _
                           DISTR_P RULES "OPTR_GROUP_4"
!
!
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
!
INTEREST_SHORTFALL GROUP 3 FULL_PREPAY      Compensate Pro_rata _
                           PARTIAL_PREPAY Compensate Pro_rata _
                           LOSS              NO_Compensate SUBORDINATED ACCUM
!
INTEREST_SHORTFALL GROUP 4 FULL_PREPAY      Compensate Pro_rata _
                           PARTIAL_PREPAY Compensate Pro_rata _
                           LOSS              NO_Compensate SUBORDINATED ACCUM
!
Tranche      Cusip      Stated Maturity
SUBS          XXXXXXXXX  20340625
3A1           XXXXXXXXX  20340625
4A1           XXXXXXXXX  20340625
!
!
CMO Block Payment Rules
-----
!

```

```

                                b04f_roll.cdi
calculate : #SubsNotGoneAgg = BBAL("SUBS") GT 0.01
!
calculate : #OrigSenPct1 = ORIG_BBAL("SNR_1") / #OrigCollBal1
!
calculate : #SenPct1 = _
                IF #SubsNotGoneAgg _
                THEN MIN(1, BBAL("SNR_1") / COLL_PREV_BAL(1)) _
                ELSE 1
!
calculate : #OrigSenPct2 = ORIG_BBAL("SNR_2") / #OrigCollBal2
!
calculate : #SenPct2 = _
                IF #SubsNotGoneAgg _
                THEN MIN(1, BBAL("SNR_2") / COLL_PREV_BAL(2)) _
                ELSE 1
!
calculate : #OrigSenPct3 = ORIG_BBAL("SNR_3") / #OrigCollBal3
!
calculate : #SenPct3 = _
                IF #SubsNotGoneAgg _
                THEN MIN(1, BBAL("SNR_3") / COLL_PREV_BAL(3)) _
                ELSE 1
!
calculate : #OrigSenPct4 = ORIG_BBAL("SNR_4") / #OrigCollBal4
!
calculate : #SenPct4 = _
                IF #SubsNotGoneAgg _
                THEN MIN(1, BBAL("SNR_4") / COLL_PREV_BAL(4)) _
                ELSE 1
!
calculate : #OrigSenPctAgg = ORIG_BBAL("SNR_1","SNR_2","SNR_3","SNR_4") /
(#OrigCollBal1 + #OrigCollBal2 + #OrigCollBal3 + #OrigCollBal4)
calculate : #SenPctAgg = MIN(1, BBAL("SNR_1","SNR_2","SNR_3","SNR_4") /
(COLL_PREV_BAL))
!
calculate : #ShiftTestAgg = NOT TRIGGER("ShiftTriggerAgg")
!
!
calculate : #Sub2TimesTriggerAgg = NOT TRIGGER("TwoTimesTriggerAgg")
calculate : #Sub2TimesTestAgg = BBAL("SUBORD_1", "SUBORD_2", "SUBORD_3",
"SUBORD_4") / BBAL("SNR_1", "SUBORD_1", "SNR_2", "SUBORD_2", "SNR_3", "SUBORD_3",
"SNR_4", "SUBORD_4") _
                                GE 2 * ORIG_BBAL("SUBORD_1", "SUBORD_2",
"SUBORD_3", "SUBORD_4")/ORIG_BBAL("SNR_1", "SUBORD_1", "SNR_2", "SUBORD_2", "SNR_3",
"SUBORD_3", "SNR_4", "SUBORD_4") AND _
                                #Sub2TimesTriggerAgg
!
calculate : #SenPctFailAgg = (#SenPctAgg > #OrigSenPctAgg)
!
calculate : #SenPrep1 = _
                IF #SenPctFailAgg _
                THEN 1 _
                ELSE #SenPct1 + SHIFT%(1) * (1-#SenPct1), _
Reduce_SHIFT%_when GROUP 1 STICKY_PASS FAILVAL_PRIOREND _
(#ShiftTestAgg)
!
calculate : #SenPrep2 = _
                IF #SenPctFailAgg _
                THEN 1 _
                ELSE #SenPct2 + SHIFT%(2) * (1-#SenPct2), _
Reduce_SHIFT%_when GROUP 2 STICKY_PASS FAILVAL_PRIOREND _
(#ShiftTestAgg)
!

```

b04f_roll.cdi

```
calculate : #SenPrep3 = _
            IF #SenPctFailAgg _
            THEN 1 _
            ELSE #SenPct3 + SHIFT%(3) * (1-#SenPct3), _
Reduce_SHIFT%_when GROUP 3 STICKY_PASS FAILVAL_PRIOREND _
(#ShiftTestAgg)
!
calculate : #SenPrep4 = _
            IF #SenPctFailAgg _
            THEN 1 _
            ELSE #SenPct4 + SHIFT%(4) * (1-#SenPct4), _
Reduce_SHIFT%_when GROUP 4 STICKY_PASS FAILVAL_PRIOREND _
(#ShiftTestAgg)
!
!
calculate : #SenPrep1 = _
            IF #SenPctFailAgg _
            THEN 1 _
            ELSE IF #Sub2TimesTestAgg _
            THEN IF CURMONTH LE 36 _
            THEN #SenPct1 + (50% * (1-#SenPct1)) _
            ELSE #SenPct1 _
            ELSE #SenPrep1
!
calculate : #SenPrep2 = _
            IF #SenPctFailAgg _
            THEN 1 _
            ELSE IF #Sub2TimesTestAgg _
            THEN IF CURMONTH LE 36 _
            THEN #SenPct2 + (50% * (1-#SenPct2)) _
            ELSE #SenPct2 _
            ELSE #SenPrep2
!
calculate : #SenPrep3 = _
            IF #SenPctFailAgg _
            THEN 1 _
            ELSE IF #Sub2TimesTestAgg _
            THEN IF CURMONTH LE 36 _
            THEN #SenPct3 + (50% * (1-#SenPct3)) _
            ELSE #SenPct3 _
            ELSE #SenPrep3
!
calculate : #SenPrep4 = _
            IF #SenPctFailAgg _
            THEN 1 _
            ELSE IF #Sub2TimesTestAgg _
            THEN IF CURMONTH LE 36 _
            THEN #SenPct4 + (50% * (1-#SenPct4)) _
            ELSE #SenPct4 _
            ELSE #SenPrep4
!
calculate : #SENRECOV1 = #SenPrep1 * DELINQ_RECOVER(1)
!
calculate : #SENRECOV2 = #SenPrep2 * DELINQ_RECOVER(2)
!
calculate : #SENRECOV3 = #SenPrep3 * DELINQ_RECOVER(3)
!
calculate : #SENRECOV4 = #SenPrep4 * DELINQ_RECOVER(4)
!
calculate: "SNR_1" _
NO_CHECK SCHEDULED      GROUP 1  FRACTION LIMIT #SCH11 = #SenPct1 , _
NO_CHECK PREPAY         GROUP 1  FRACTION LIMIT #PRP11 = #SenPrep1 , _
NO_CHECK RECOVER        GROUP 1  AMOUNT   LIMIT #REC11 = #SENRECOV1
```

b04f_roll.cdi

```

!
calculate: "SNR_2" -
NO_CHECK SCHEDULED - GROUP 2 FRACTION LIMIT #SCH22 = #SenPct2 , -
NO_CHECK PREPAY GROUP 2 FRACTION LIMIT #PRP22 = #SenPrep2 , -
NO_CHECK RECOVER GROUP 2 AMOUNT LIMIT #REC22 = #SENRECOV2
!
calculate: "SNR_3" -
NO_CHECK SCHEDULED - GROUP 3 FRACTION LIMIT #SCH33 = #SenPct3 , -
NO_CHECK PREPAY GROUP 3 FRACTION LIMIT #PRP33 = #SenPrep3 , -
NO_CHECK RECOVER GROUP 3 AMOUNT LIMIT #REC33 = #SENRECOV3
!
calculate: "SNR_4" -
NO_CHECK SCHEDULED - GROUP 4 FRACTION LIMIT #SCH44 = #SenPct4 , -
NO_CHECK PREPAY GROUP 4 FRACTION LIMIT #PRP44 = #SenPrep4 , -
NO_CHECK RECOVER GROUP 4 AMOUNT LIMIT #REC44 = #SENRECOV4
!
calculate : #SenSchedAlloc1 = #SCH11 * COLL_P_SCHED(1)
calculate : #SenPrepayAlloc1 = #PRP11 * COLL_P_PREPAY(1)
calculate : #SenRecoverAlloc1 = #REC11
calculate : #SenSchedAlloc2 = #SCH22 * COLL_P_SCHED(2)
calculate : #SenPrepayAlloc2 = #PRP22 * COLL_P_PREPAY(2)
calculate : #SenRecoverAlloc2 = #REC22
calculate : #SenSchedAlloc3 = #SCH33 * COLL_P_SCHED(3)
calculate : #SenPrepayAlloc3 = #PRP33 * COLL_P_PREPAY(3)
calculate : #SenRecoverAlloc3 = #REC33
calculate : #SenSchedAlloc4 = #SCH44 * COLL_P_SCHED(4)
calculate : #SenPrepayAlloc4 = #PRP44 * COLL_P_PREPAY(4)
calculate : #SenRecoverAlloc4 = #REC44
!
!! ***** CROSS SENIORS DUE TO RAPID PREPAYS *****
*****
!
calculate : #SubMultipleAgg = LOOKUP_TBL( "STEP", CURMONTH ,
"CROSSPREP_2TIMESAgg", "MONTH", "MULTIPLE" )
!
calculate : #CrossTestAgg = ((#SubBal1 + #SubBal2 + #SubBal3 + #SubBal4) /
(COLL_PREV_BAL(1) + COLL_PREV_BAL(2) + COLL_PREV_BAL(3) + COLL_PREV_BAL(4)) LT _
#SubMultipleAgg * (#OrigSubBal1 + #OrigSubBal2 + #OrigSubBal3 +
#OrigSubBal4) / (#OrigCollBal1 + #OrigCollBal2 + #OrigCollBal3 + #OrigCollBal4))
!
calculate : #CrossTest1 = _
BBAL("SNR_1") LT 0.01 AND #CrossTestAgg
!
calculate : #CrossTest2 = _
BBAL("SNR_2") LT 0.01 AND #CrossTestAgg
!
calculate : #CrossTest3 = _
BBAL("SNR_3") LT 0.01 AND #CrossTestAgg
!
calculate : #CrossTest4 = _
BBAL("SNR_4") LT 0.01 AND #CrossTestAgg
!
calculate: "SNR_1" -
SCHEDULED GROUP 2 AMOUNT LIMIT #CROSS2SCH1 = IF #CrossTest2 THEN
(COLL_P_SCHED(2) - #SenSchedAlloc2) * BBAL("SNR_1")/BBAL("SNR_1","SNR_3","SNR_4")
ELSE 0 ; -
PREPAY GROUP 2 AMOUNT LIMIT #CROSS2PRP1 = IF #CrossTest2 THEN
(COLL_P_PREPAY(2) - #SenPrepayAlloc2) * BBAL("SNR_1")/BBAL("SNR_1","SNR_3","SNR_4")
ELSE 0 ; -
RECOVER GROUP 2 AMOUNT LIMIT #CROSS2REC1 = IF #CrossTest2 THEN
(DELINQ_RECOVER(2) - #SenRecoverAlloc2) *
BBAL("SNR_1")/BBAL("SNR_1","SNR_3","SNR_4") ELSE 0 ; -
SCHEDULED GROUP 3 AMOUNT LIMIT #CROSS3SCH1 = IF #CrossTest3 THEN

```

```

                                b04f_roll.cdi
(COLL_P_SCHED(3) - #SenSchedAlloc3) * BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_4")
ELSE 0 ; -
PREPAY GROUP 3 AMOUNT LIMIT #CROSS3PRP1 = IF #CrossTest3 THEN
(COLL_P_PREPAY(3) - #SenPrepayAlloc3) * BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_4")
ELSE 0 ; -
RECOVER GROUP 3 AMOUNT LIMIT #CROSS3REC1 = IF #CrossTest3 THEN
(DELINQ_RECOVER(3) - #SenRecoverAlloc3) *
BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_4") ELSE 0 ; -
SCHEDULED GROUP 4 AMOUNT LIMIT #CROSS4SCH1 = IF #CrossTest4 THEN
(COLL_P_SCHED(4) - #SenSchedAlloc4) * BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; -
PREPAY GROUP 4 AMOUNT LIMIT #CROSS4PRP1 = IF #CrossTest4 THEN
(COLL_P_PREPAY(4) - #SenPrepayAlloc4) * BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; -
RECOVER GROUP 4 AMOUNT LIMIT #CROSS4REC1 = IF #CrossTest4 THEN
(DELINQ_RECOVER(4) - #SenRecoverAlloc4) *
BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_3") ELSE 0
!
calculate: "SNR_2" -
SCHEDULED GROUP 1 AMOUNT LIMIT #CROSS1SCH2 = IF #CrossTest1 THEN
(COLL_P_SCHED(1) - #SenSchedAlloc1) * BBAL("SNR_2")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
PREPAY GROUP 1 AMOUNT LIMIT #CROSS1PRP2 = IF #CrossTest1 THEN
(COLL_P_PREPAY(1) - #SenPrepayAlloc1) * BBAL("SNR_2")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
RECOVER GROUP 1 AMOUNT LIMIT #CROSS1REC2 = IF #CrossTest1 THEN
(DELINQ_RECOVER(1) - #SenRecoverAlloc1) *
BBAL("SNR_2")/BBAL("SNR_2","SNR_3","SNR_4") ELSE 0 ; -
SCHEDULED GROUP 3 AMOUNT LIMIT #CROSS3SCH2 = IF #CrossTest3 THEN
(COLL_P_SCHED(3) - #SenSchedAlloc3) * BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_4")
ELSE 0 ; -
PREPAY GROUP 3 AMOUNT LIMIT #CROSS3PRP2 = IF #CrossTest3 THEN
(COLL_P_PREPAY(3) - #SenPrepayAlloc3) * BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_4")
ELSE 0 ; -
RECOVER GROUP 3 AMOUNT LIMIT #CROSS3REC2 = IF #CrossTest3 THEN
(DELINQ_RECOVER(3) - #SenRecoverAlloc3) *
BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_4") ELSE 0 ; -
SCHEDULED GROUP 4 AMOUNT LIMIT #CROSS4SCH2 = IF #CrossTest4 THEN
(COLL_P_SCHED(4) - #SenSchedAlloc4) * BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; -
PREPAY GROUP 4 AMOUNT LIMIT #CROSS4PRP2 = IF #CrossTest4 THEN
(COLL_P_PREPAY(4) - #SenPrepayAlloc4) * BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; -
RECOVER GROUP 4 AMOUNT LIMIT #CROSS4REC2 = IF #CrossTest4 THEN
(DELINQ_RECOVER(4) - #SenRecoverAlloc4) *
BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_3") ELSE 0
!
calculate: "SNR_3" -
SCHEDULED GROUP 1 AMOUNT LIMIT #CROSS1SCH3 = IF #CrossTest1 THEN
(COLL_P_SCHED(1) - #SenSchedAlloc1) * BBAL("SNR_3")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
PREPAY GROUP 1 AMOUNT LIMIT #CROSS1PRP3 = IF #CrossTest1 THEN
(COLL_P_PREPAY(1) - #SenPrepayAlloc1) * BBAL("SNR_3")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
RECOVER GROUP 1 AMOUNT LIMIT #CROSS1REC3 = IF #CrossTest1 THEN
(DELINQ_RECOVER(1) - #SenRecoverAlloc1) *
BBAL("SNR_3")/BBAL("SNR_2","SNR_3","SNR_4") ELSE 0 ; -
SCHEDULED GROUP 2 AMOUNT LIMIT #CROSS2SCH3 = IF #CrossTest2 THEN
(COLL_P_SCHED(2) - #SenSchedAlloc2) * BBAL("SNR_3")/BBAL("SNR_1","SNR_3","SNR_4")
ELSE 0 ; -
PREPAY GROUP 2 AMOUNT LIMIT #CROSS2PRP3 = IF #CrossTest2 THEN
(COLL_P_PREPAY(2) - #SenPrepayAlloc2) * BBAL("SNR_3")/BBAL("SNR_1","SNR_3","SNR_4")
ELSE 0 ; -

```

```

                                b04f_roll.cdi
RECOVER      GROUP 2  AMOUNT LIMIT #CROSS2REC3 = IF #CrossTest2 THEN
(DELINQ_RECOVER(2) - #SenRecoverAlloc2) *
BBAL("SNR_3")/BBAL("SNR_1","SNR_3","SNR_4") ELSE 0 ; -
SCHEDULED    GROUP 4  AMOUNT LIMIT #CROSS4SCH3 = IF #CrossTest4 THEN
(COLL_P_SCHED(4) - #SenSchedAlloc4) * BBAL("SNR_3")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; -
PREPAY       GROUP 4  AMOUNT LIMIT #CROSS4PRP3 = IF #CrossTest4 THEN
(COLL_P_PREPAY(4) - #SenPrepayAlloc4) * BBAL("SNR_3")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; -
RECOVER      GROUP 4  AMOUNT LIMIT #CROSS4REC3 = IF #CrossTest4 THEN
(DELINQ_RECOVER(4) - #SenRecoverAlloc4) *
BBAL("SNR_3")/BBAL("SNR_1","SNR_2","SNR_3") ELSE 0
!
calculate: "SNR_4" -
SCHEDULED    GROUP 1  AMOUNT LIMIT #CROSS1SCH4 = IF #CrossTest1 THEN
(COLL_P_SCHED(1) - #SenSchedAlloc1) * BBAL("SNR_4")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
PREPAY       GROUP 1  AMOUNT LIMIT #CROSS1PRP4 = IF #CrossTest1 THEN
(COLL_P_PREPAY(1) - #SenPrepayAlloc1) * BBAL("SNR_4")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
RECOVER      GROUP 1  AMOUNT LIMIT #CROSS1REC4 = IF #CrossTest1 THEN
(DELINQ_RECOVER(1) - #SenRecoverAlloc1) *
BBAL("SNR_4")/BBAL("SNR_2","SNR_3","SNR_4") ELSE 0 ; -
SCHEDULED    GROUP 2  AMOUNT LIMIT #CROSS2SCH4 = IF #CrossTest2 THEN
(COLL_P_SCHED(2) - #SenSchedAlloc2) * BBAL("SNR_4")/BBAL("SNR_1","SNR_3","SNR_4")
ELSE 0 ; -
PREPAY       GROUP 2  AMOUNT LIMIT #CROSS2PRP4 = IF #CrossTest2 THEN
(COLL_P_PREPAY(2) - #SenPrepayAlloc2) * BBAL("SNR_4")/BBAL("SNR_1","SNR_3","SNR_4")
ELSE 0 ; -
RECOVER      GROUP 2  AMOUNT LIMIT #CROSS2REC4 = IF #CrossTest2 THEN
(DELINQ_RECOVER(2) - #SenRecoverAlloc2) *
BBAL("SNR_4")/BBAL("SNR_1","SNR_3","SNR_4") ELSE 0 ; -
SCHEDULED    GROUP 3  AMOUNT LIMIT #CROSS3SCH4 = IF #CrossTest3 THEN
(COLL_P_SCHED(3) - #SenSchedAlloc3) * BBAL("SNR_4")/BBAL("SNR_1","SNR_2","SNR_4")
ELSE 0 ; -
PREPAY       GROUP 3  AMOUNT LIMIT #CROSS3PRP4 = IF #CrossTest3 THEN
(COLL_P_PREPAY(3) - #SenPrepayAlloc3) * BBAL("SNR_4")/BBAL("SNR_1","SNR_2","SNR_4")
ELSE 0 ; -
RECOVER      GROUP 3  AMOUNT LIMIT #CROSS3REC4 = IF #CrossTest3 THEN
(DELINQ_RECOVER(3) - #SenRecoverAlloc3) *
BBAL("SNR_4")/BBAL("SNR_1","SNR_2","SNR_4") ELSE 0
!
calculate : #SenSchedAlloc1 = #SenSchedAlloc1 + #CROSS1SCH2 + #CROSS1SCH3 +
#CROSS1SCH4
calculate : #SenPrepayAlloc1 = #SenPrepayAlloc1 + #CROSS1PRP2 + #CROSS1PRP3 +
#CROSS1PRP4
calculate : #SenRecoverAlloc1 = #SenRecoverAlloc1 + #CROSS1REC2 + #CROSS1REC3 +
#CROSS1REC4
!
calculate : #SenSchedAlloc2 = #SenSchedAlloc2 + #CROSS2SCH1 + #CROSS2SCH3 +
#CROSS2SCH4
calculate : #SenPrepayAlloc2 = #SenPrepayAlloc2 + #CROSS2PRP1 + #CROSS2PRP3 +
#CROSS2PRP4
calculate : #SenRecoverAlloc2 = #SenRecoverAlloc2 + #CROSS2REC1 + #CROSS2REC3 +
#CROSS2REC4
!
calculate : #SenSchedAlloc3 = #SenSchedAlloc3 + #CROSS3SCH1 + #CROSS3SCH2 +
#CROSS3SCH4
calculate : #SenPrepayAlloc3 = #SenPrepayAlloc3 + #CROSS3PRP1 + #CROSS3PRP2 +
#CROSS3PRP4
calculate : #SenRecoverAlloc3 = #SenRecoverAlloc3 + #CROSS3REC1 + #CROSS3REC2 +
#CROSS3REC4
!

```

```

                                b04f_roll.cdi
calculate : #SenSchedAlloc4 = #SenSchedAlloc4 + #CROSS4SCH1 + #CROSS4SCH2 +
#CROSS4SCH3
calculate : #SenPrepayAlloc4 = #SenPrepayAlloc4 + #CROSS4PRP1 + #CROSS4PRP2 +
#CROSS4PRP3
calculate : #SenRecoverAlloc4 = #SenRecoverAlloc4 + #CROSS4REC1 + #CROSS4REC2 +
#CROSS4REC3
!
!! ***** END CROSS SENIORS DUE TO RAPID PREPAYS
*****
!
calculate : #SubSched1 = MAX( 0, COLL_P_SCHED(1) - #SenSchedAlloc1 )
calculate : #SubPrepay1 = MAX( 0, COLL_P_PREPAY(1) - #SenPrepayAlloc1 )
calculate : #SubRecov1 = MAX( 0, DELINQ_RECOVER(1) - #SenRecoverAlloc1 )
!
calculate : #SubSched2 = MAX( 0, COLL_P_SCHED(2) - #SenSchedAlloc2 )
calculate : #SubPrepay2 = MAX( 0, COLL_P_PREPAY(2) - #SenPrepayAlloc2 )
calculate : #SubRecov2 = MAX( 0, DELINQ_RECOVER(2) - #SenRecoverAlloc2 )
!
calculate : #SubSched3 = MAX( 0, COLL_P_SCHED(3) - #SenSchedAlloc3 )
calculate : #SubPrepay3 = MAX( 0, COLL_P_PREPAY(3) - #SenPrepayAlloc3 )
calculate : #SubRecov3 = MAX( 0, DELINQ_RECOVER(3) - #SenRecoverAlloc3 )
!
calculate : #SubSched4 = MAX( 0, COLL_P_SCHED(4) - #SenSchedAlloc4 )
calculate : #SubPrepay4 = MAX( 0, COLL_P_PREPAY(4) - #SenPrepayAlloc4 )
calculate : #SubRecov4 = MAX( 0, DELINQ_RECOVER(4) - #SenRecoverAlloc4 )
!
calculate: "SUBORD_1" _
NO_CHECK SCHEDULED      GROUP 1  AMOUNT      = #SubSched1 , _
NO_CHECK PREPAY         GROUP 1  AMOUNT      = #SubPrepay1 , _
NO_CHECK RECOVER        GROUP 1  AMOUNT      = #SubRecov1
!
calculate: "SUBORD_2" _
NO_CHECK SCHEDULED      GROUP 2  AMOUNT      = #SubSched2 , _
NO_CHECK PREPAY         GROUP 2  AMOUNT      = #SubPrepay2 , _
NO_CHECK RECOVER        GROUP 2  AMOUNT      = #SubRecov2
!
calculate: "SUBORD_3" _
NO_CHECK SCHEDULED      GROUP 3  AMOUNT      = #SubSched3 , _
NO_CHECK PREPAY         GROUP 3  AMOUNT      = #SubPrepay3 , _
NO_CHECK RECOVER        GROUP 3  AMOUNT      = #SubRecov3
!
calculate: "SUBORD_4" _
NO_CHECK SCHEDULED      GROUP 4  AMOUNT      = #SubSched4 , _
NO_CHECK PREPAY         GROUP 4  AMOUNT      = #SubPrepay4 , _
NO_CHECK RECOVER        GROUP 4  AMOUNT      = #SubRecov4
!
calculate : #SubSched = #SubSched1 + #SubSched2 + #SubSched3 + #SubSched4
calculate : #SubPrepay = #SubPrepay1 + #SubPrepay2 + #SubPrepay3 + #SubPrepay4
calculate : #SubRecov = #SubRecov1 + #SubRecov2 + #SubRecov3 + #SubRecov4
!
calculate: "SUBS" _
NO_CHECK SCHEDULED      AMOUNT      = #SubSched , _
NO_CHECK PREPAY         AMOUNT      = #SubPrepay , _
NO_CHECK RECOVER        AMOUNT      = #SubRecov
!
calculate : #ExcessSub = EXCESS_SUB_AMT
!
calculate : #ExcessSub = EXCESS_SUB_AMT("SUBS")
!
calculate : #ExcessSub1 = #ExcessSub * BBAL("SNR_1") /
BBAL("SNR_1","SNR_2","SNR_3","SNR_4")
calculate : #ExcessSub2 = #ExcessSub * BBAL("SNR_2") /
BBAL("SNR_1","SNR_2","SNR_3","SNR_4")

```

```

                                b04f_rol1.cdi
    calculate : #ExcessSub3      = #ExcessSub * BBAL("SNR_3") /
BBAL("SNR_1","SNR_2","SNR_3","SNR_4")
    calculate : #ExcessSub4      = #ExcessSub * BBAL("SNR_4") /
BBAL("SNR_1","SNR_2","SNR_3","SNR_4")
!
    calculate : "SNR_1" _
    NO_CHECK EXCESS_SUB_AMT AMOUNT LIMIT #VS1 = #ExcessSub1
!
    calculate : "SNR_2" _
    NO_CHECK EXCESS_SUB_AMT AMOUNT LIMIT #VS2 = #ExcessSub2
!
    calculate : "SNR_3" _
    NO_CHECK EXCESS_SUB_AMT AMOUNT LIMIT #VS3 = #ExcessSub3
!
    calculate : "SNR_4" _
    NO_CHECK EXCESS_SUB_AMT AMOUNT LIMIT #VS4 = #ExcessSub4
!
    calculate : #SubExcessSub1    = MIN(OPTIMAL_PRINCPMT("SUBORD_1"), #VS1)
    calculate : #RemSubExcessSub1 = #VS1 - #SubExcessSub1
    calculate : #RemSubOptPrinc1  = OPTIMAL_PRINCPMT("SUBORD_1") - #SubExcessSub1
    calculate : #SubExcessSub2    = MIN(OPTIMAL_PRINCPMT("SUBORD_2"), #VS2)
    calculate : #RemSubExcessSub2 = #VS2 - #SubExcessSub2
    calculate : #RemSubOptPrinc2  = OPTIMAL_PRINCPMT("SUBORD_2") - #SubExcessSub2
    calculate : #SubExcessSub3    = MIN(OPTIMAL_PRINCPMT("SUBORD_3"), #VS3)
    calculate : #RemSubExcessSub3 = #VS3 - #SubExcessSub3
    calculate : #RemSubOptPrinc3  = OPTIMAL_PRINCPMT("SUBORD_3") - #SubExcessSub3
    calculate : #SubExcessSub4    = MIN(OPTIMAL_PRINCPMT("SUBORD_4"), #VS4)
    calculate : #RemSubExcessSub4 = #VS4 - #SubExcessSub4
    calculate : #RemSubOptPrinc4  = OPTIMAL_PRINCPMT("SUBORD_4") - #SubExcessSub4
    calculate : #RemSubExcessSub  = #RemSubExcessSub1 + #RemSubExcessSub2 +
#RemSubExcessSub3 + #RemSubExcessSub4
    calculate : #RemSubOptPrinc   = #RemSubOptPrinc1 + #RemSubOptPrinc2 +
#RemSubOptPrinc3 + #RemSubOptPrinc4
!
    calculate : #TotSubExcessSub1 = -1 * (#SubExcessSub1 + #RemSubExcessSub *
#RemSubOptPrinc1/#RemSubOptPrinc)
    calculate : #TotSubExcessSub2 = -1 * (#SubExcessSub2 + #RemSubExcessSub *
#RemSubOptPrinc2/#RemSubOptPrinc)
    calculate : #TotSubExcessSub3 = -1 * (#SubExcessSub3 + #RemSubExcessSub *
#RemSubOptPrinc3/#RemSubOptPrinc)
    calculate : #TotSubExcessSub4 = -1 * (#SubExcessSub4 + #RemSubExcessSub *
#RemSubOptPrinc4/#RemSubOptPrinc)
!
    calculate : "SUBORD_1" _
    NO_CHECK NEG_OK CUSTOM AMOUNT          = #TotSubExcessSub1
!
    calculate : "SUBORD_2" _
    NO_CHECK NEG_OK CUSTOM AMOUNT          = #TotSubExcessSub2
!
    calculate : "SUBORD_3" _
    NO_CHECK NEG_OK CUSTOM AMOUNT          = #TotSubExcessSub3
!
    calculate : "SUBORD_4" _
    NO_CHECK NEG_OK CUSTOM AMOUNT          = #TotSubExcessSub4
!
-----
    pay : CLASS INTEREST PRO_RATE ("SNR_1" )
    pay : CLASS INTSHORT PRO_RATE ("SNR_1" )
    pay : CLASS PRINCIPAL SEQUENTIAL ( "SNR_1" )
-----
    pay : CLASS INTEREST PRO_RATE ("SNR_2" )
    pay : CLASS INTSHORT PRO_RATE ("SNR_2" )
    pay : CLASS PRINCIPAL SEQUENTIAL ( "SNR_2" )

```

b04f_roll.cdi

```

-----
pay : CLASS INTEREST PRO_RATE ( "SNR_3" )
pay : CLASS INTSHORT PRO_RATE ( "SNR_3" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "SNR_3" )
-----
pay : CLASS INTEREST PRO_RATE ( "SNR_4" )
pay : CLASS INTSHORT PRO_RATE ( "SNR_4" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "SNR_4" )
-----
pay : CLASS INTEREST PRO_RATE ( "SUBORD_1" )
pay : CLASS INTSHORT PRO_RATE ( "SUBORD_1" )
-----
pay : CLASS INTEREST PRO_RATE ( "SUBORD_2" )
pay : CLASS INTSHORT PRO_RATE ( "SUBORD_2" )
-----
pay : CLASS INTEREST PRO_RATE ( "SUBORD_3" )
pay : CLASS INTSHORT PRO_RATE ( "SUBORD_3" )
-----
pay : CLASS INTEREST PRO_RATE ( "SUBORD_4" )
pay : CLASS INTSHORT PRO_RATE ( "SUBORD_4" )
-----
from : CLASS ( "GRP1"; "GRP2"; "GRP3"; "GRP4" )
pay : CLASS INTEREST PRO_RATE ( "SUBORD_1"; "SUBORD_2"; "SUBORD_3";
"SUBORD_4" )
pay : CLASS INTSHORT PRO_RATE ( "SUBORD_1"; "SUBORD_2"; "SUBORD_3";
"SUBORD_4" )
-----
from : CLASS ( "GRP1" ; "GRP2" ; "GRP3" ; "GRP4" )
pay : CLASS ENTIRETY PRO_RATE ( "SNR_1" ; "SNR_2" ; "SNR_3" ; "SNR_4" )
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "SUBORD_1" )
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "SUBORD_2" )
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "SUBORD_3" )
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "SUBORD_4" )
-----
from : CLASS ( "GRP1"; "GRP2"; "GRP3"; "GRP4" )
pay : CLASS PRINCIPAL PRO_RATE ( "SUBORD_1"; "SUBORD_2"; "SUBORD_3";
"SUBORD_4" )
-----
from : CLASS ( "SNR_1" )
pay : CLASS INTEREST PRO_RATE ( "RESID"; "1A1" )
pay : CLASS INTSHORT PRO_RATE ( "RESID"; "1A1" )
-----
from : CLASS ( "SNR_1" )
pay : CLASS BALANCE SEQUENTIAL ( "RESID", "1A1" )
-----
!
-----
from : CLASS ( "RESID" )
pay : SEQUENTIAL ( "RESID#1" )
-----
from : CLASS ( "1A1" )
pay : SEQUENTIAL ( "1A1#1" )
-----
subject to : CEILING ( (BBAL("SUBORD_1#1")-BBAL("SUBORD_1"))) )
pay : SEQUENTIAL ( "SUBORD_1#1" )
-----
!
from : CLASS ( "SNR_2" )
pay : CLASS INTEREST PRO_RATE ( "SEQ"; "PT"; "2WAC-IO" )

```

```

                                b04f_roll.cdi
pay : CLASS INTSHORT PRO_RATA ( "SEQ"; "PT"; "2WAC-IO" )
-----
from : CLASS ( "SNR_2" )
pay : CLASS BALANCE PRO_RATA ( "SEQ" ; "PT" )
-----
!
from : CLASS ( "SEQ" )
pay : CLASS INTEREST PRO_RATA ( "2A1"; "2A2"; "2A3"; "2A4"; "2A5"; "2A6"
)
pay : CLASS INTSHORT PRO_RATA ( "2A1"; "2A2"; "2A3"; "2A4"; "2A5"; "2A6"
)
-----
from : CLASS ( "SEQ" )
pay : CLASS BALANCE SEQUENTIAL ( "2A1", "2A2", "2A3", "2A4", "2A5", "2A6"
)
-----
!
from : CLASS ( "PT" )
pay : CLASS INTEREST PRO_RATA ( "2A7" )
pay : CLASS INTSHORT PRO_RATA ( "2A7" )
-----
from : CLASS ( "PT" )
pay : CLASS BALANCE SEQUENTIAL ( "2A7" )
-----
!
-----
from : CLASS ( "2A1" )
pay : SEQUENTIAL ( "2A1#1" )
-----
from : CLASS ( "2A2" )
pay : SEQUENTIAL ( "2A2#1" )
-----
from : CLASS ( "2A3" )
pay : SEQUENTIAL ( "2A3#1" )
-----
from : CLASS ( "2A4" )
pay : SEQUENTIAL ( "2A4#1" )
-----
from : CLASS ( "2A5" )
pay : SEQUENTIAL ( "2A5#1" )
-----
from : CLASS ( "2A6" )
pay : SEQUENTIAL ( "2A6#1" )
-----
from : CLASS ( "2A7" )
pay : SEQUENTIAL ( "2A7#1" )
-----
subject to : CEILING ( (BBAL("SUBORD_2#1")-BBAL("SUBORD_2"))) )
pay : SEQUENTIAL ( "SUBORD_2#1" )
-----
!
from : CLASS ( "SNR_3" )
pay : CLASS INTEREST PRO_RATA ( "3A1" )
pay : CLASS INTSHORT PRO_RATA ( "3A1" )
-----
from : CLASS ( "SNR_3" )
pay : CLASS BALANCE SEQUENTIAL ( "3A1" )
-----
!
-----
from : CLASS ( "3A1" )
pay : SEQUENTIAL ( "3A1#1" )
-----

```

```

                                b04f_roll.cdi
subject to : CEILING ( (BBAL("SUBORD_3#1")-BBAL("SUBORD_3"))) )
pay : SEQUENTIAL ( "SUBORD_3#1" )
-----
!
from : CLASS ( "SNR_4" )
pay : CLASS INTEREST PRO_RATA ( "4A1" )
pay : CLASS INTSHORT PRO_RATA ( "4A1" )
-----
from : CLASS ( "SNR_4" )
pay : CLASS BALANCE SEQUENTIAL ( "4A1" )
-----
!
-----
from : CLASS ( "4A1" )
pay : SEQUENTIAL ( "4A1#1" )
-----
subject to : CEILING ( (BBAL("SUBORD_4#1")-BBAL("SUBORD_4"))) )
pay : SEQUENTIAL ( "SUBORD_4#1" )
-----
calculate : #P_SUBS =
PRINCPMT("SUBORD_1#1","SUBORD_2#1","SUBORD_3#1","SUBORD_4#1")
-----
from : CLASS ( "SUBORD_1" ; "SUBORD_2" ; "SUBORD_3" ; "SUBORD_4" )
pay : CLASS INTEREST PRO_RATA ( "SUBS" )
pay : CLASS INTSHORT PRO_RATA ( "SUBS" )
-----
from : CLASS ( "SUBORD_1" ; "SUBORD_2" ; "SUBORD_3" ; "SUBORD_4" )
subject to : CEILING ( #P_SUBS )
pay : CLASS BALANCE SEQUENTIAL ( "SUBS" )
-----
from : CLASS ( "SUBORD_1" ; "SUBORD_2" ; "SUBORD_3" ; "SUBORD_4" )
pay : CLASS MORE_INTEREST SEQUENTIAL ( "GRP1" )
-----
!
-----
from : CLASS ( "SUBS" )
pay : SEQUENTIAL ( "SUBS#1" )
-----
!
-----
PAYDOWN SUBORD TRANCHES
calculate : #PrincReduce = BBAL("SUBS#1") - BBAL("SUBORD_1", "SUBORD_2",
"SUBORD_3", "SUBORD_4")
calculate : #SubPrinc1 = BBAL("SUBORD_1#1") - BBAL("SUBORD_1")
calculate : #SubPrinc2 = BBAL("SUBORD_2#1") - BBAL("SUBORD_2")
calculate : #SubPrinc3 = BBAL("SUBORD_3#1") - BBAL("SUBORD_3")
calculate : #SubPrinc4 = BBAL("SUBORD_4#1") - BBAL("SUBORD_4")
calculate : #SubPrincAgg = #SubPrinc1 + #SubPrinc2 + #SubPrinc3 + #SubPrinc4
calculate : #PrincReduce1 = #PrincReduce * #SubPrinc1 / #SubPrincAgg
calculate : #PrincReduce2 = #PrincReduce * #SubPrinc2 / #SubPrincAgg
calculate : #PrincReduce3 = #PrincReduce * #SubPrinc3 / #SubPrincAgg
calculate : #PrincReduce4 = #PrincReduce * #SubPrinc4 / #SubPrincAgg
-----
subject to : CEILING ( ( #SubPrinc1 - #PrincReduce1 ) )
pay : SEQUENTIAL ( "SUBORD_1#1" )
-----
subject to : CEILING ( ( #SubPrinc2 - #PrincReduce2 ) )
pay : SEQUENTIAL ( "SUBORD_2#1" )
-----
subject to : CEILING ( ( #SubPrinc3 - #PrincReduce3 ) )
pay : SEQUENTIAL ( "SUBORD_3#1" )
-----
subject to : CEILING ( ( #SubPrinc4 - #PrincReduce4 ) )

```

```

                                b04f_roll.cdi
pay : SEQUENTIAL ( "SUBORD_4#1" )
-----
!
----- MANUAL WRITEDOWNS
calculate : #ReduceSubord1 = MAX(0, MIN( BBAL("SUBORD_1#1"), BBAL( "RESID#1",
"1A1#1", "SUBORD_1#1" ) - COLL_BAL(1)))
calculate : #ReduceSubord2 = MAX(0, MIN( BBAL("SUBORD_2#1"), BBAL( "2A1#1",
"2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1" ) - COLL_BAL(2)))
calculate : #ReduceSubord3 = MAX(0, MIN( BBAL("SUBORD_3#1"), BBAL( "3A1#1",
"SUBORD_3#1" ) - COLL_BAL(3)))
calculate : #ReduceSubord4 = MAX(0, MIN( BBAL("SUBORD_4#1"), BBAL( "4A1#1",
"SUBORD_4#1" ) - COLL_BAL(4)))
calculate : #TotReduceSubord = #ReduceSubord1 + #ReduceSubord2 + #ReduceSubord3
+ #ReduceSubord4
-----
when : IS_THERE ( "SUBS#1" )
pay : DECREMENT ( BALANCE "SUBORD_1#1", BY #ReduceSubord1 )
-----
calculate : #IncreaseSubord1 = MAX(0, MIN( #TotReduceSubord, COLL_BAL(1) -
BBAL( "RESID#1", "1A1#1", "SUBORD_1#1" )))
-----
when : IS_THERE ( "SUBS#1" )
pay : INCREMENT ( BALANCE "SUBORD_1#1", BY #IncreaseSubord1 )
-----
when : IS_THERE ( "SUBS#1" )
pay : DECREMENT ( BALANCE "SUBORD_2#1", BY #ReduceSubord2 )
-----
calculate : #IncreaseSubord2 = MAX(0, MIN( #TotReduceSubord, COLL_BAL(2) -
BBAL( "2A1#1", "2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1"
)))
-----
when : IS_THERE ( "SUBS#1" )
pay : INCREMENT ( BALANCE "SUBORD_2#1", BY #IncreaseSubord2 )
-----
when : IS_THERE ( "SUBS#1" )
pay : DECREMENT ( BALANCE "SUBORD_3#1", BY #ReduceSubord3 )
-----
calculate : #IncreaseSubord3 = MAX(0, MIN( #TotReduceSubord, COLL_BAL(3) -
BBAL( "3A1#1", "SUBORD_3#1" )))
-----
when : IS_THERE ( "SUBS#1" )
pay : INCREMENT ( BALANCE "SUBORD_3#1", BY #IncreaseSubord3 )
-----
when : IS_THERE ( "SUBS#1" )
pay : DECREMENT ( BALANCE "SUBORD_4#1", BY #ReduceSubord4 )
-----
calculate : #IncreaseSubord4 = MAX(0, MIN( #TotReduceSubord, COLL_BAL(4) -
BBAL( "4A1#1", "SUBORD_4#1" )))
-----
when : IS_THERE ( "SUBS#1" )
pay : INCREMENT ( BALANCE "SUBORD_4#1", BY #IncreaseSubord4 )
-----
calculate : #MoreReduceSubord1 = MAX(0, MIN( BBAL("SUBORD_1#1"),
BBAL("SUBORD_1#1") / BBAL("SUBORD_1#1", "SUBORD_2#1", "SUBORD_3#1", "SUBORD_4#1") *
( BBAL("RESID#1", "1A1#1", "SUBORD_1#1", "2A1#1", "2A2#1", "2A3#1", "2A4#1",
"2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1", "3A1#1", "SUBORD_3#1", "4A1#1",
"SUBORD_4#1") - COLL_BAL(1)))
calculate : #MoreReduceSubord2 = MAX(0, MIN( BBAL("SUBORD_2#1"),
BBAL("SUBORD_2#1") / BBAL("SUBORD_1#1", "SUBORD_2#1", "SUBORD_3#1", "SUBORD_4#1") *
( BBAL("RESID#1", "1A1#1", "SUBORD_1#1", "2A1#1", "2A2#1", "2A3#1", "2A4#1",
"2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1", "3A1#1", "SUBORD_3#1", "4A1#1",
"SUBORD_4#1") - COLL_BAL(2)))
calculate : #MoreReduceSubord3 = MAX(0, MIN( BBAL("SUBORD_3#1"),

```

```

                                b04f_roll.cdi
BBAL("SUBORD_3#1") / BBAL("SUBORD_1#1", "SUBORD_2#1", "SUBORD_3#1", "SUBORD_4#1") *
( BBAL("RESID#1", "1A1#1", "SUBORD_1#1", "2A1#1", "2A2#1", "2A3#1", "2A4#1",
"2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1", "3A1#1", "SUBORD_3#1", "4A1#1",
"SUBORD_4#1") - COLL_BAL)))
    calculate : #MoreReduceSubord4 = MAX(0, MIN( BBAL("SUBORD_4#1"),
BBAL("SUBORD_4#1") / BBAL("SUBORD_1#1", "SUBORD_2#1", "SUBORD_3#1", "SUBORD_4#1") *
( BBAL("RESID#1", "1A1#1", "SUBORD_1#1", "2A1#1", "2A2#1", "2A3#1", "2A4#1",
"2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1", "3A1#1", "SUBORD_3#1", "4A1#1",
"SUBORD_4#1") - COLL_BAL)))
-----
    when : IS_TRUE ( (BBAL("SUBS#1") GT 0.01 ) AND (( BBAL("SUBORD_2#1") LT
0.01 ) OR ( BBAL("SUBORD_3#1") LT 0.01 ) OR ( BBAL("SUBORD_4#1") LT 0.01 ) ))
    pay : DECREMENT ( BALANCE "SUBORD_1#1", BY #MoreReduceSubord1 )
-----
    when : IS_TRUE ( (BBAL("SUBS#1") GT 0.01 ) AND (( BBAL("SUBORD_1#1") LT
0.01 ) OR ( BBAL("SUBORD_3#1") LT 0.01 ) OR ( BBAL("SUBORD_4#1") LT 0.01 ) ))
    pay : DECREMENT ( BALANCE "SUBORD_2#1", BY #MoreReduceSubord2 )
-----
    when : IS_TRUE ( (BBAL("SUBS#1") GT 0.01 ) AND (( BBAL("SUBORD_1#1") LT
0.01 ) OR ( BBAL("SUBORD_2#1") LT 0.01 ) OR ( BBAL("SUBORD_4#1") LT 0.01 ) ))
    pay : DECREMENT ( BALANCE "SUBORD_3#1", BY #MoreReduceSubord3 )
-----
    when : IS_TRUE ( (BBAL("SUBS#1") GT 0.01 ) AND (( BBAL("SUBORD_1#1") LT
0.01 ) OR ( BBAL("SUBORD_2#1") LT 0.01 ) OR ( BBAL("SUBORD_3#1") LT 0.01 ) ))
    pay : DECREMENT ( BALANCE "SUBORD_4#1", BY #MoreReduceSubord4 )
-----
    calculate : #WriteDown = MAX(0, BBAL( "RESID#1", "1A1#1", "SUBS#1", "2A1#1",
"2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1", "3A1#1", "4A1#1" ) -
COLL_BAL(1,2,3,4))
-----
    from : SUBACCOUNT ( #WriteDown )
    pay : WRITEDOWN PRO_RATA ( "SUBS#1" )
-----
    calculate : #SenWriteDown1 = MAX(0, BBAL( "RESID#1", "1A1#1" ) / BBAL(
"RESID#1", "1A1#1", "2A1#1", "2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1",
"3A1#1", "4A1#1" ) * #WriteDown)
-----
    calculate : #SenWriteDown2 = MAX(0, BBAL( "2A1#1", "2A2#1", "2A3#1", "2A4#1",
"2A5#1", "2A6#1", "2A7#1" ) / BBAL( "RESID#1", "1A1#1", "2A1#1", "2A2#1", "2A3#1",
"2A4#1", "2A5#1", "2A6#1", "2A7#1", "3A1#1", "4A1#1" ) * #WriteDown)
-----
    calculate : #SenWriteDown3 = MAX(0, BBAL( "3A1#1" ) / BBAL( "RESID#1", "1A1#1",
"2A1#1", "2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1", "3A1#1", "4A1#1" ) *
#WriteDown)
-----
    calculate : #SenWriteDown4 = MAX(0, BBAL( "4A1#1" ) / BBAL( "RESID#1", "1A1#1",
"2A1#1", "2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1", "3A1#1", "4A1#1" ) *
#WriteDown)
-----
    from : SUBACCOUNT ( #WriteDown, #SenWriteDown1 )
    pay : WRITEDOWN PRO_RATA ( "RESID#1"; "1A1#1" )
-----
    from : SUBACCOUNT ( #WriteDown, #SenWriteDown2 )
    pay : WRITEDOWN PRO_RATA ( "2A1#1"; "2A2#1"; "2A3#1"; "2A4#1"; "2A5#1";
"2A6#1"; "2A7#1" )
-----
    from : SUBACCOUNT ( #WriteDown, #SenWriteDown3 )
    pay : WRITEDOWN SEQUENTIAL ( "3A1#1" )
-----
    from : SUBACCOUNT ( #WriteDown, #SenWriteDown4 )
    pay : WRITEDOWN SEQUENTIAL ( "4A1#1" )
-----

```

!

```

                                b04f_roll.cdi
calculate : #BondBal1      = BBAL("RESID#1","1A1#1","SUBORD_1#1")
-----
calculate : #BondBal2      =
BBAL("2A1#1","2A2#1","2A3#1","2A4#1","2A5#1","2A6#1","2A7#1","SUBORD_2#1")
-----
calculate : #BondBal3      = BBAL("3A1#1","SUBORD_3#1")
-----
calculate : #BondBal4      = BBAL("4A1#1","SUBORD_4#1")
-----
calculate : #BondBal       = #BondBal1 + #BondBal2 + #BondBal3 + #BondBal4
-----
    when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (INTPMT("2A1#1")) )
    pay : INTEREST SEQUENTIAL ("GMC_2A1#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A1#1")) )
    pay : SEQUENTIAL ("GMC_2A1#1")
-----
calculate : #GMCWD = BBAL("GMC_2A1#1") - BBAL("2A1#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A1#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A1#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("GMC_2A1#1")) )
    pay : SEQUENTIAL ("GMC_2A1#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A1#1") * COUPON("2A1#1") / 1200 * 0 / 30) )
    pay : AS_INTEREST ("GMC_2A1#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (INTPMT("2A2#1")) )
    pay : INTEREST SEQUENTIAL ("GMC_2A2#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A2#1")) )
    pay : SEQUENTIAL ("GMC_2A2#1")
-----
calculate : #GMCWD = BBAL("GMC_2A2#1") - BBAL("2A2#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A2#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A2#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("GMC_2A2#1")) )

```

```

                                b04f_roll.cdi
pay : SEQUENTIAL ("GMC_2A2#1")
-----
when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A2#1") * COUPON("2A2#1") / 1200 * 0 / 30) )
pay : AS_INTEREST ("GMC_2A2#1")
-----
when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (INTPMT("2A3#1")) )
pay : INTEREST SEQUENTIAL ("GMC_2A3#1")
-----
when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A3#1")) )
pay : SEQUENTIAL ("GMC_2A3#1")
-----
calculate : #GMCWD = BBAL("GMC_2A3#1") - BBAL("2A3#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A3#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A3#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A3#1")) )
    pay : SEQUENTIAL ("GMC_2A3#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A3#1") * COUPON("2A3#1") / 1200 * 24 / 30) )
    pay : AS_INTEREST ("GMC_2A3#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (INTPMT("2A4#1")) )
    pay : INTEREST SEQUENTIAL ("GMC_2A4#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A4#1")) )
    pay : SEQUENTIAL ("GMC_2A4#1")
-----
    calculate : #GMCWD = BBAL("GMC_2A4#1") - BBAL("2A4#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A4#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A4#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A4#1")) )
    pay : SEQUENTIAL ("GMC_2A4#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A4#1") * COUPON("2A4#1") / 1200 * 24 / 30) )
    pay : AS_INTEREST ("GMC_2A4#1")

```

b04f_roll.cdi

```
-----
when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (INTPMT("2A5#1")) )
pay : INTEREST SEQUENTIAL ("GMC_2A5#1")
-----
when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A5#1")) )
pay : SEQUENTIAL ("GMC_2A5#1")
-----
calculate : #GMCWD = BBAL("GMC_2A5#1") - BBAL("2A5#1")
-----
ifdef #cmover_3.0f _
when : IS_TRUE ( CURMONTH LE 59 )
from : SUBACCOUNT ( #GMCWD )
pay : WRITEDOWN SEQUENTIAL ("GMC_2A5#1")
-----
!
ifndef #cmover_3.0f _
when : IS_TRUE ( CURMONTH LE 59 )
pay : DECREMENT ( BALANCE "GMC_2A5#1", BY #GMCWD )
-----
!
when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("GMC_2A5#1")) )
pay : SEQUENTIAL ("GMC_2A5#1")
-----
when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A5#1") * COUPON("2A5#1") / 1200 * 24 / 30) )
pay : AS_INTEREST ("GMC_2A5#1")
-----
when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (INTPMT("2A6#1")) )
pay : INTEREST SEQUENTIAL ("GMC_2A6#1")
-----
when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A6#1")) )
pay : SEQUENTIAL ("GMC_2A6#1")
-----
calculate : #GMCWD = BBAL("GMC_2A6#1") - BBAL("2A6#1")
-----
ifdef #cmover_3.0f _
when : IS_TRUE ( CURMONTH LE 59 )
from : SUBACCOUNT ( #GMCWD )
pay : WRITEDOWN SEQUENTIAL ("GMC_2A6#1")
-----
!
ifndef #cmover_3.0f _
when : IS_TRUE ( CURMONTH LE 59 )
pay : DECREMENT ( BALANCE "GMC_2A6#1", BY #GMCWD )
-----
!
when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("GMC_2A6#1")) )
pay : SEQUENTIAL ("GMC_2A6#1")
-----
when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A6#1") * COUPON("2A6#1") / 1200 * 24 / 30) )
pay : AS_INTEREST ("GMC_2A6#1")
-----
calculate: #CallBalGrp1 = COLL_BAL(1)
calculate: #CallBalGrp2 = COLL_BAL(2)
calculate: #CallBalGrp3 = COLL_BAL(3)
```

```

                                b04f_roll.cdi
calculate: #CallBalGrp4 = COLL_BAL(4)
----- 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 : SEQUENTIAL ( "RESID#1", "1A1#1" )
-----
      from : CLASS ( "GRP1" )
      pay : CLASS BALANCE SEQUENTIAL ( "SUBORD_1" )
-----
subject to : CEILING ( (BBAL("SUBORD_1#1")-BBAL("SUBORD_1"))) )
      pay : SEQUENTIAL ( "SUBORD_1#1" )
-----
      from : CLASS ( "SUBORD_1" )
      pay : SEQUENTIAL ( "SUBS#1" )
-----
calculate : #WriteDown1 = BBAL("GRP1")
-----
calculate : #WD_SUBORD_1 = MIN(BBAL("SUBORD_1#1"), #Writedown1)
calculate : #Writedown1 = #Writedown1 - #WD_SUBORD_1
-----
subject to : CEILING ( #WD_SUBORD_1 )
      pay : DECREMENT( BALANCE "SUBORD_1#1", BY #WD_SUBORD_1 )
-----
      from : SUBACCOUNT ( #Writedown1 )
      pay : WRITEDOWN PRO_RATE ( "RESID#1"; "1A1#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 : SEQUENTIAL ( "2A1#1", "2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1",
"2A7#1" )
-----
      from : CLASS ( "GRP2" )
      pay : CLASS BALANCE SEQUENTIAL ( "SUBORD_2" )
-----
subject to : CEILING ( (BBAL("SUBORD_2#1")-BBAL("SUBORD_2"))) )
      pay : SEQUENTIAL ( "SUBORD_2#1" )
-----
      from : CLASS ( "SUBORD_2" )
      pay : SEQUENTIAL ( "SUBS#1" )
-----
      when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A1#1"))) )
      pay : SEQUENTIAL ( "GMC_2A1#1" )
-----
calculate : #GMCWD = BBAL("GMC_2A1#1") - BBAL("2A1#1")
-----
ifdef #cmover_3.0f _
      when : IS_TRUE ( CURMONTH LE 59 )
      from : SUBACCOUNT ( #GMCWD )
      pay : WRITEDOWN SEQUENTIAL ( "GMC_2A1#1" )
-----

```

b04f_roll.cdi

```
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A1#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A1#1")) )
    pay : SEQUENTIAL ("GMC_2A1#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A1#1") * COUPON("2A1#1") / 1200 * 0 / 30) )
    pay : AS_INTEREST ("GMC_2A1#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A2#1")) )
    pay : SEQUENTIAL ("GMC_2A2#1")
-----
    calculate : #GMCWD = BBAL("GMC_2A2#1") - BBAL("2A2#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A2#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A2#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A2#1")) )
    pay : SEQUENTIAL ("GMC_2A2#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A2#1") * COUPON("2A2#1") / 1200 * 0 / 30) )
    pay : AS_INTEREST ("GMC_2A2#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A3#1")) )
    pay : SEQUENTIAL ("GMC_2A3#1")
-----
    calculate : #GMCWD = BBAL("GMC_2A3#1") - BBAL("2A3#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A3#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A3#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A3#1")) )
    pay : SEQUENTIAL ("GMC_2A3#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A3#1") * COUPON("2A3#1") / 1200 * 24 / 30) )
    pay : AS_INTEREST ("GMC_2A3#1")
```

b04f_roll.cdi

```
-----
      when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A4#1")) )
      pay : SEQUENTIAL ("GMC_2A4#1")
-----
      calculate : #GMCWD = BBAL("GMC_2A4#1") - BBAL("2A4#1")
-----
ifdef #cmover_3.0f _
      when : IS_TRUE ( CURMONTH LE 59 )
      from : SUBACCOUNT ( #GMCWD )
      pay : WRITEDOWN SEQUENTIAL ("GMC_2A4#1")
-----
!
ifndef #cmover_3.0f _
      when : IS_TRUE ( CURMONTH LE 59 )
      pay : DECREMENT ( BALANCE "GMC_2A4#1", BY #GMCWD )
-----
!
      when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("GMC_2A4#1")) )
      pay : SEQUENTIAL ("GMC_2A4#1")
-----
      when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A4#1") * COUPON("2A4#1") / 1200 * 24 / 30) )
      pay : AS_INTEREST ("GMC_2A4#1")
-----
      when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A5#1")) )
      pay : SEQUENTIAL ("GMC_2A5#1")
-----
      calculate : #GMCWD = BBAL("GMC_2A5#1") - BBAL("2A5#1")
-----
ifdef #cmover_3.0f _
      when : IS_TRUE ( CURMONTH LE 59 )
      from : SUBACCOUNT ( #GMCWD )
      pay : WRITEDOWN SEQUENTIAL ("GMC_2A5#1")
-----
!
ifndef #cmover_3.0f _
      when : IS_TRUE ( CURMONTH LE 59 )
      pay : DECREMENT ( BALANCE "GMC_2A5#1", BY #GMCWD )
-----
!
      when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("GMC_2A5#1")) )
      pay : SEQUENTIAL ("GMC_2A5#1")
-----
      when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A5#1") * COUPON("2A5#1") / 1200 * 24 / 30) )
      pay : AS_INTEREST ("GMC_2A5#1")
-----
      when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A6#1")) )
      pay : SEQUENTIAL ("GMC_2A6#1")
-----
      calculate : #GMCWD = BBAL("GMC_2A6#1") - BBAL("2A6#1")
-----
ifdef #cmover_3.0f _
      when : IS_TRUE ( CURMONTH LE 59 )
      from : SUBACCOUNT ( #GMCWD )
      pay : WRITEDOWN SEQUENTIAL ("GMC_2A6#1")
-----
!
```

b04f_roll.cdi

```
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A6#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A6#1")) )
    pay : SEQUENTIAL ("GMC_2A6#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A6#1") * COUPON("2A6#1") / 1200 * 24 / 30) )
    pay : AS_INTEREST ("GMC_2A6#1")
-----
    calculate : #WriteDown2 = BBAL("GRP2")
-----
    calculate : #WD_SUBORD_2 = MIN(BBAL("SUBORD_2#1"), #writedown2)
    calculate : #writedown2 = #writedown2 - #WD_SUBORD_2
-----
    subject to : CEILING ( #WD_SUBORD_2 )
    pay : DECREMENT( BALANCE "SUBORD_2#1", BY #WD_SUBORD_2 )
-----
    from : SUBACCOUNT ( #writedown2 )
    pay : WRITEDOWN PRO_RATA ( "2A1#1"; "2A2#1"; "2A3#1"; "2A4#1"; "2A5#1";
"2A6#1"; "2A7#1" )
-----
----- SECTION: "OPTR_GROUP_3"
-----
    from : CASH_ACCOUNT (100)
    subject to : CEILING ( #CallBalGrp3)
    pay : CLASS BALANCE SEQUENTIAL ( "GRP3" )
    pay : CLASS MORE_INTEREST SEQUENTIAL ( "GRP3" )
-----
    from : CLASS ( "GRP3" )
    pay : SEQUENTIAL ( "3A1#1" )
-----
    from : CLASS ( "GRP3" )
    pay : CLASS BALANCE SEQUENTIAL ( "SUBORD_3" )
-----
    subject to : CEILING ( (BBAL("SUBORD_3#1")-BBAL("SUBORD_3")) )
    pay : SEQUENTIAL ( "SUBORD_3#1" )
-----
    from : CLASS ( "SUBORD_3" )
    pay : SEQUENTIAL ( "SUBS#1" )
-----
    calculate : #WriteDown3 = BBAL("GRP3")
-----
    calculate : #WD_SUBORD_3 = MIN(BBAL("SUBORD_3#1"), #writedown3)
    calculate : #writedown3 = #writedown3 - #WD_SUBORD_3
-----
    subject to : CEILING ( #WD_SUBORD_3 )
    pay : DECREMENT( BALANCE "SUBORD_3#1", BY #WD_SUBORD_3 )
-----
    from : SUBACCOUNT ( #writedown3 )
    pay : WRITEDOWN SEQUENTIAL ( "3A1#1" )
-----
----- SECTION: "OPTR_GROUP_4"
-----
    from : CASH_ACCOUNT (100)
    subject to : CEILING ( #CallBalGrp4)
    pay : CLASS BALANCE SEQUENTIAL ( "GRP4" )
    pay : CLASS MORE_INTEREST SEQUENTIAL ( "GRP4" )
-----
    from : CLASS ( "GRP4" )
```

```

                                b04f_roll.cdi
pay : SEQUENTIAL ( "4A1#1" )
-----
from : CLASS ( "GRP4" )
pay : CLASS BALANCE SEQUENTIAL ( "SUBORD_4" )
-----
subject to : CEILING ( (BBAL("SUBORD_4#1")-BBAL("SUBORD_4")) )
pay : SEQUENTIAL ( "SUBORD_4#1" )
-----
from : CLASS ( "SUBORD_4" )
pay : SEQUENTIAL ( "SUBS#1" )
-----
calculate : #Writedown4 = BBAL("GRP4")
-----
calculate : #WD_SUBORD_4 = MIN(BBAL("SUBORD_4#1"), #Writedown4)
calculate : #Writedown4 = #Writedown4 - #WD_SUBORD_4
-----
subject to : CEILING ( #WD_SUBORD_4 )
pay : DECREMENT( BALANCE "SUBORD_4#1", BY #WD_SUBORD_4 )
-----
from : SUBACCOUNT ( #Writedown4 )
pay : WRITEDOWN SEQUENTIAL ( "4A1#1" )
-----

```

```

!
Schedule "SHIFT1%"

```

```

Declare
SHIFTINT GROUP 1
84      100%
96      70%
108     60%
120     40%
132     20%
144     0%

```

```

!
!
Schedule "SHIFT2%"

```

```

Declare
SHIFTINT GROUP 2
84      100%
96      70%
108     60%
120     40%
132     20%
144     0%

```

```

!
!
Schedule "SHIFT3%"

```

```

Declare
SHIFTINT GROUP 3
84      100%
96      70%
108     60%
120     40%
132     20%
144     0%

```

```

!
!
Schedule "SHIFT4%"

```

```

Declare
SHIFTINT GROUP 4
84      100%
96      70%
108     60%
120     40%

```

b04f_roll.cdi

132 20%
144 0%

Collateral

! Factor --Delay--
! Type Date P/Y BV Use BV for 0
! WL 20040601 9999 9999 FALSE

! Pool# Type Gross Current Original --Fee-- Maturity Orig ARM
! Gross #mos #mos P#mos P#mos Life Reset Life Max Look
! Coupon Factor Balance P/Y BV P/Y BV Term Index
! Margin ToRst RstPer ToRst RstPer Cap Cap Floor Negam Back

!! BEGINNING OF COLLATERAL

M 1 "1" WL 00 WAC 4.5444382129 (1066826.43 /
1066826.43); 1066826.43 0.3750000000 0.3750000000
352:8 352:8 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
29 12 SYNC_INT 10.5444382129 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 4.5444382129

GROUP 1 TEASER
M 2 "1" WL 00 WAC 4.5000000000 (362485.72 /
362485.72); 362485.72 0.3750000000 0.3750000000
354:6 354:6 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
31 12 SYNC_INT 10.5000000000 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 4.5000000000

GROUP 1 TEASER
M 3 "1" WL 00 WAC 4.7500000000 (615866.29 /
615866.29); 615866.29 0.3750000000 0.3750000000
356:4 356:4 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
33 12 SYNC_INT 10.7500000000 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 4.7500000000

GROUP 1 TEASER
M 4 "1" WL 00 WAC 3.9665503634 (6375274.01 /
6375274.01); 6375274.01 0.3750000000 0.3750000000
358:2 358:2 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
35 12 SYNC_INT 9.9665503634 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 3.9665503634

GROUP 1 TEASER
M 5 "1" WL 00 WAC 4.0298692227 (165409098.04 /
165409098.04); 165409098.04 0.3750000000 0.3750000000
358:1 358:1 359 NO_CHECK ARM LIBOR_1YR 2.2500000000
36 12 SYNC_INT 10.0298692227 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 4.0298692227

GROUP 1 TEASER
M 6 "1" WL 00 WAC 4.4452795773 (55537325.37 /
55537325.37); 55537325.37 0.3750000000 0.3750000000
360:0 360:0 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
37 12 SYNC_INT 10.4452795773 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 4.4452795773

GROUP 1 TEASER
M 7 "2" WL 00 WAC 5.0000000000 (596992.32 /
596992.32); 596992.32 0.2500000000 0.2500000000
347:13 347:13 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
48 12 SYNC_INT 10.0000000000 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 5.0000000000

GROUP 2 TEASER
M 8 "2" WL 00 WAC 4.5428934028 (1097688.53 /
1097688.53); 1097688.53 0.2500000000 0.2500000000
351:9 351:9 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
52 12 SYNC_INT 9.5428934028 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.5428934028
GROUP 2 TEASER

```

                                b04f_roll.cdi
M          9      "2"      WL      00      WAC      4.1926161867 (      3471014.97 /
3471014.97 );      3471014.97      0.2500000000 0.2500000000
      309:8      309:8      317 NO_CHECK ARM LIBOR_1YR      2.2500000000
53  12 SYNC_INT      9.1926161867      2.0000000000      2.2500000000      0      0
INIT_PERCAP      5.0000000000 ORIG_GROSSRATE 4.1926161867
GROUP 2      TEASER
M          10      "2"      WL      00      WAC      4.8560092813 (      822898.16 /
822898.16 );      822898.16      0.2500000000 0.2500000000
      354:6      354:6      360 NO_CHECK ARM LIBOR_1YR      2.2500000000
55  12 SYNC_INT      9.8560092813      2.0000000000      2.2500000000      0      0
INIT_PERCAP      5.0000000000 ORIG_GROSSRATE 4.8560092813
GROUP 2      TEASER
M          11      "2"      WL      00      WAC      4.5531677162 (      4037666.05 /
4037666.05 );      4037666.05      0.2500000000 0.2500000000
      355:5      355:5      360 NO_CHECK ARM LIBOR_1YR      2.2500000000
56  12 SYNC_INT      9.5531677162      2.0000000000      2.2500000000      0      0
INIT_PERCAP      5.0000000000 ORIG_GROSSRATE 4.5531677162
GROUP 2      TEASER
M          12      "2"      WL      00      WAC      4.8750000000 (      337322.48 /
337322.48 );      337322.48      0.2500000000 0.2500000000
      356:4      356:4      360 NO_CHECK ARM LIBOR_1YR      2.2500000000
57  12 SYNC_INT      9.8750000000      2.0000000000      2.2500000000      0      0
INIT_PERCAP      5.0000000000 ORIG_GROSSRATE 4.8750000000
GROUP 2      TEASER
M          13      "2"      WL      00      WAC      4.8003425711 (      1113206.23 /
1113206.23 );      1113206.23      0.2500000000 0.2500000000
      357:3      357:3      360 NO_CHECK ARM LIBOR_1YR      2.2500000000
58  12 SYNC_INT      9.8003425711      2.0000000000      2.2500000000      0      0
INIT_PERCAP      5.0000000000 ORIG_GROSSRATE 4.8003425711
GROUP 2      TEASER
M          14      "2"      WL      00      WAC      4.3982787548 (      24277616.88 /
24277616.88 );      24277616.88      0.2500000000 0.2500000000
      353:2      353:2      355 NO_CHECK ARM LIBOR_1YR      2.2500000000
59  12 SYNC_INT      9.3982787548      2.0000000000      2.2500000000      0      0
INIT_PERCAP      5.0000000000 ORIG_GROSSRATE 4.3982787548
GROUP 2      TEASER
M          15      "2"      WL      00      WAC      4.3685480805 (      348613931.93 /
348613931.93 );      348613931.93      0.2500000000 0.2500000000
      353:1      353:1      354 NO_CHECK ARM LIBOR_1YR      2.2500000000
60  12 SYNC_INT      9.3685480805      2.0000000000      2.2500000000      0      0
INIT_PERCAP      5.0000000000 ORIG_GROSSRATE 4.3685480805
GROUP 2      TEASER
M          16      "2"      WL      00      WAC      4.6153520857 (      84169387.12 /
84169387.12 );      84169387.12      0.2500000000 0.2500000000
      356:0      356:0      356 NO_CHECK ARM LIBOR_1YR      2.2500000000
61  12 SYNC_INT      9.6153520857      2.0000000000      2.2500000000      0      0
INIT_PERCAP      5.0000000000 ORIG_GROSSRATE 4.6153520857
GROUP 2      TEASER
M          17      "2-IO"      WL      00      WAC      4.6250000000 (      390400.00 /
390400.00 );      390400.00      0.2500000000 0.2500000000
      355:5      355:5      360 NO_CHECK ARM LIBOR_1YR      2.2500000000
56  12 SYNC_INT      9.6250000000      2.0000000000      2.2500000000      0      0
INIT_PERCAP      5.0000000000 ORIG_GROSSRATE 4.6250000000 AMORT NONE FOR      60
GROUP 2      TEASER
M          18      "2-IO"      WL      00      WAC      5.3750000000 (      333701.00 /
333701.00 );      333701.00      0.2500000000 0.2500000000
      356:4      356:4      360 NO_CHECK ARM LIBOR_1YR      2.2500000000
57  12 SYNC_INT      10.3750000000      2.0000000000      2.2500000000      0      0
INIT_PERCAP      5.0000000000 ORIG_GROSSRATE 5.3750000000 AMORT NONE FOR      60
GROUP 2      TEASER
M          19      "2-IO"      WL      00      WAC      4.2342266461 (      3091257.60 /
3091257.60 );      3091257.60      0.2500000000 0.2500000000
      358:2      358:2      360 NO_CHECK ARM LIBOR_1YR      2.2500000000

```

b04f_roll.cdi

```

59 12 SYNC_INT      9.2342266461  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.2342266461 AMORT NONE FOR      60
GROUP 2 TEASER
M 20 "2-IO" WL 00 WAC 4.3970325002 ( 450419045.56 /
450419045.56 ); 450419045.56 0.2500000000 0.2500000000
359:1 359:1 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
60 12 SYNC_INT      9.3970325002  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.3970325002 AMORT NONE FOR      60
GROUP 2 TEASER
M 21 "2-IO" WL 00 WAC 4.6341836373 ( 168511754.13 /
168511754.13 ); 168511754.13 0.2500000000 0.2500000000
360:0 360:0 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
61 12 SYNC_INT      9.6341836373  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.6341836373 AMORT NONE FOR      60
GROUP 2 TEASER
M 22 "3" WL 00 WAC 5.7500000000 ( 135661.59 /
135661.59 ); 135661.59 0.2500000000 0.2500000000
167:13 167:13 180 NO_CHECK ARM LIBOR_1YR 2.2500000000
48 12 SYNC_INT      10.7500000000  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 5.7500000000
GROUP 3 TEASER
M 23 "3" WL 00 WAC 5.1250000000 ( 212163.31 /
212163.31 ); 212163.31 0.2500000000 0.2500000000
349:11 349:11 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
50 12 SYNC_INT      10.1250000000  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 5.1250000000
GROUP 3 TEASER
M 24 "3" WL 00 WAC 4.7500000000 ( 268887.51 /
268887.51 ); 268887.51 0.2500000000 0.2500000000
352:8 352:8 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
53 12 SYNC_INT      9.7500000000  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.7500000000
GROUP 3 TEASER
M 25 "3" WL 00 WAC 5.5000000000 ( 320193.22 /
320193.22 ); 320193.22 0.2500000000 0.2500000000
353:7 353:7 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
54 12 SYNC_INT      10.5000000000  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 5.5000000000
GROUP 3 TEASER
M 26 "3" WL 00 WAC 4.5000000000 ( 147805.10 /
147805.10 ); 147805.10 0.2500000000 0.2500000000
359:1 359:1 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
60 12 SYNC_INT      9.5000000000  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.5000000000
GROUP 3 TEASER
M 27 "3-IO" WL 00 WAC 4.8750000000 ( 226000.00 /
226000.00 ); 226000.00 0.2500000000 0.2500000000
354:6 354:6 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
55 12 SYNC_INT      9.8750000000  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.8750000000 AMORT NONE FOR      60
GROUP 3 TEASER
M 28 "3-IO" WL 00 WAC 4.4582975399 ( 3582237.53 /
3582237.53 ); 3582237.53 0.2500000000 0.2500000000
357:3 357:3 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
58 12 SYNC_INT      9.4582975399  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.4582975399 AMORT NONE FOR      60
GROUP 3 TEASER
M 29 "3-IO" WL 00 WAC 4.3663695038 ( 9233678.39 /
9233678.39 ); 9233678.39 0.2500000000 0.2500000000
358:2 358:2 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
59 12 SYNC_INT      9.3663695038  2.0000000000  2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.3663695038 AMORT NONE FOR      60
GROUP 3 TEASER

```

b04f_roll.cdi

M	30	"3-IO"	WL	00	WAC	4.4423488603 (	150127639.25 /	
150127639.25);		150127639.25				0.2500000000	0.2500000000	
	359:1	359:1		360	NO_CHECK ARM LIBOR_1YR		2.2500000000	
60	12	SYNC_INT		9.4423488603	2.0000000000	2.2500000000	0	0
INIT_PERCAP		5.0000000000	ORIG_GROSSRATE	4.4423488603	AMORT NONE	FOR		60
GROUP 3		TEASER						
M	31	"3-IO"	WL	00	WAC	4.7355598691 (	51027081.71 /	
51027081.71);		51027081.71				0.2500000000	0.2500000000	
	360:0	360:0		360	NO_CHECK ARM LIBOR_1YR		2.2500000000	
61	12	SYNC_INT		9.7355598691	2.0000000000	2.2500000000	0	0
INIT_PERCAP		5.0000000000	ORIG_GROSSRATE	4.7355598691	AMORT NONE	FOR		60
GROUP 3		TEASER						
M	32	"4"	WL	00	WAC	4.9017954660 (	5971807.97 /	
5971807.97);		5971807.97				0.2500000000	0.2500000000	
	358:2	358:2		360	NO_CHECK ARM LIBOR_1YR		2.2500000000	
83	12	SYNC_INT		9.9017954660	2.0000000000	2.2500000000	0	0
INIT_PERCAP		5.0000000000	ORIG_GROSSRATE	4.9017954660				
GROUP 4		TEASER						
M	33	"4"	WL	00	WAC	4.8481000167 (	84798305.31 /	
84798305.31);		84798305.31				0.2500000000	0.2500000000	
	355:1	355:1		356	NO_CHECK ARM LIBOR_1YR		2.2500000000	
84	12	SYNC_INT		9.8481000167	2.0000000000	2.2500000000	0	0
INIT_PERCAP		5.0000000000	ORIG_GROSSRATE	4.8481000167				
GROUP 4		TEASER						
M	34	"4"	WL	00	WAC	5.1939228940 (	18286720.91 /	
18286720.91);		18286720.91				0.2500000000	0.2500000000	
	360:0	360:0		360	NO_CHECK ARM LIBOR_1YR		2.2500000000	
85	12	SYNC_INT		10.1939228940	2.0000000000	2.2500000000	0	0
INIT_PERCAP		5.0000000000	ORIG_GROSSRATE	5.1939228940				
GROUP 4		TEASER						

b04f_call.cdi
! B04F_CALL.CDI #CMOVER_3.0E WHOLE_LOAN ! MAX_CF_VECTSIZE 620

!! Created by Intex Deal Maker v3.7.011 , subroutines 3.1
!! 06/06/2004 4:36 PM

! Modeled in the Intex CMO Modeling Language, (B000BDB448D23)
! 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.

!The tables and other statistical analyses (the "Hypothetical Performance Data")
that you will produce using Intex with the attached information are privileged and
intended solely for use by you

!(the party to whom Banc of America Securities LLC provided the computer file source
code used to generate them). The Hypothetical Performance Data will be generated by
you using a computer file source

!code prepared by Banc of America Securities LLC in reliance upon information
furnished by the issuer of the securities and its affiliates, the accuracy and
completeness of which has not been verified by

!Banc of America Securities LLC or any other person. The computer file source code
that you will use to prepare the Hypothetical Performance Data was furnished to you
solely by Banc of America Securities LLC

!and not by the issuer of the securities. It may not be (a) used for any purpose
other than to make a preliminary evaluation of the referenced securities or (b)
provided by you to any third party other than your

!legal, tax, financial and/or accounting advisors for the purposes of evaluating the
Hypothetical Performance Data. You agree that the Hypothetical Performance Data
will be generated by or on behalf of you,

!and that neither Banc of America Securities LLC nor anyone acting on its behalf has
generated or is in any way responsible for any Hypothetical Performance Data.

!Numerous assumptions were used in preparing the computer file source code you will
use to generate the Hypothetical Performance Data. Those assumptions may or may not
be reflected in the Hypothetical

!Performance Data. As such, no assurance can be given as to the Hypothetical
Performance Data's accuracy, appropriateness or completeness in any particular
context; nor as to whether the Hypothetical

!Performance Data and/or the assumptions upon which it is based reflect present
market conditions or future market performance. The Hypothetical Performance Data
should not be construed as either projections

!or predictions or as legal, tax, financial or accounting advice.

!Any weighted average lives, yields and principal payment periods shown in the
Hypothetical Performance Data will be based on prepayment assumptions, and changes
in such prepayment assumptions may

!dramatically affect such weighted average lives, yields and principal payment
periods. In addition, it is possible that prepayments on the underlying assets will
occur at rates slower or faster than the rates

!shown in the Hypothetical Performance Data. Furthermore, unless otherwise
provided, the Hypothetical Performance Data assumes no losses on the underlying
assets and no interest shortfall. The specific

!characteristics of the securities may differ from those shown in the Hypothetical
Performance Data due to, among other things, differences between (a) the actual
underlying assets and the hypothetical

!underlying assets used in preparing the Hypothetical Performance Data and (b) the
assumptions used by you in producing the Hypothetical Performance Data and the
actual assumptions used in pricing the

!actual securities. The principal amount, designation and terms of any security
described in the Hypothetical Performance Data are subject to change prior to
issuance. You should contact the Banc of America

!Securities LLC Trading Desk at (704) 388-1579 to confirm the final principal
amount, designation and terms of any security described in this communication prior
to committing to purchase that security. Neither

b04f_call.cdi

!Banc of America Securities LLC nor any of its affiliates makes any representation or warranty as to the actual rate or timing of payments on any of the underlying assets or the payments or yield on the securities.

!Although a registration statement (including a prospectus) relating to the securities discussed in this communication has been filed with the Securities and Exchange Commission and is effective, the final

!prospectus supplement relating to the securities discussed in this communication has not yet been filed with the Securities and Exchange Commission. This communication shall not constitute an offer to sell

!or the solicitation of an offer to buy nor shall there be any sale of the securities discussed in this communication in any state in which such offer, solicitation or sale would be unlawful prior to registration or

!qualification of such securities under the securities laws of any such state. The principal amount, designation and terms of any security described in the computer model and Hypothetical Performance Data are

!preliminary and subject to change prior to issuance.

!Prospective purchasers are referred to the final prospectus supplement relating to the securities discussed in this communication for definitive yield and maturity information regarding those securities, based

!on the final principal amounts, designations and terms of those securities. Once available, a final prospectus and prospectus supplement may be obtained by contacting the Banc of America Securities LLC

!Trading Desk at (704) 388-1579.

!The computer model referenced herein supersedes all computer models related to the subject securities that have been made available to you previously. In addition, this computer model will be superseded in

!its entirety by the final prospectus supplement relating to the actual securities preliminarily described by this computer model.

!Please be advised that the securities described herein may not be appropriate for all investors. Potential investors must be willing to assume, among other things, market price volatility, prepayment, yield curve

!and interest rate risks. Investors should make every effort to consider the risks of these securities.

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

COLLAT_GROUPS 1 2 3 4

!
!

```
DEFINE CONSTANT #OrigCollBa1 = 1644988940.62
DEFINE CONSTANT #OrigCollBa11 = 229366875.86
DEFINE CONSTANT #OrigCollBa12 = 1091283882.96
DEFINE CONSTANT #OrigCollBa13 = 215281347.61
DEFINE CONSTANT #OrigCollBa14 = 109056834.19
```

!

```
DEFINE CONSTANT #OrigBondBa1 = 1644988940.62
DEFINE CONSTANT #OrigBondBa11 = 229366875.86
DEFINE CONSTANT #OrigBondBa12 = 1091283882.96
DEFINE CONSTANT #OrigBondBa13 = 215281347.61
DEFINE CONSTANT #OrigBondBa14 = 109056834.19
```

!

```
DEFINE #BondBa11 = 271128940.62
DEFINE #BondBa12 = 1107619840.62
DEFINE #BondBa13 = 257458840.62
DEFINE #BondBa14 = 154367840.62
DEFINE #BondBa1 = 1790575462.48
```

!

FULL_DEALNAME: b04f_call

!

```

b04f_call.cdi
DEAL SIZE:                $ 1644988940.62
PRICING SPEED:            20% CPR
ISSUE DATE:              20040601
SETTLEMENT DATE:         20040628

Record date delay:       24

DEFINE TR_INDEXDEPS_ALL

DEFINE SCHEDULE "SHIFT1%", "SHIFT2%", "SHIFT3%", "SHIFT4%"

TRUSTEE_FEE    GROUP 1      0.002
TRUSTEE_FEE    GROUP 2      0.002
TRUSTEE_FEE    GROUP 3      0.002
TRUSTEE_FEE    GROUP 4      0.002

DEAL_CLOCK_INFO _
ISSUE_CDU_DATE      20040601 _
DEAL_FIRSTPAY_DATE  20040725 _

DEFINE DYNAMIC STICKY #NetRate = ( COLL_I_MISC("COUPON") ) / COLL_PREV_BAL * 1200
DEFINE DYNAMIC STICKY #NetRate1 = ( COLL_I_MISC("COUPON",1) ) / COLL_PREV_BAL(1)
* 1200
DEFINE DYNAMIC STICKY #NetRate2 = ( COLL_I_MISC("COUPON",2) ) / COLL_PREV_BAL(2)
* 1200
DEFINE DYNAMIC STICKY #NetRate3 = ( COLL_I_MISC("COUPON",3) ) / COLL_PREV_BAL(3)
* 1200
DEFINE DYNAMIC STICKY #NetRate4 = ( COLL_I_MISC("COUPON",4) ) / COLL_PREV_BAL(4)
* 1200

DEFINE TABLE "SI_LOSSAagg" (6, 2) = "MONTH" "SHIFTR"
36.1    20%
96.1    30%
108.1   35%
120.1   40%
132.1   45%
144.1   50%

DEFINE TABLE "SI_2TIMESLOSSAgg" (6, 2) = "MONTH" "PCT"
36.1    20%
96.1    30%
108.1   35%
120.1   40%
132.1   45%
144.1   50%

DEFINE TABLE "CROSSPREP_2TIMESAgg" (1, 2) = "MONTH" "MULTIPLE"
360.1    2

DEFINE #COUPON_SUBS = 0

TOLERANCE CLEANUP 0.00
TOLERANCE WRITEDOWN_OLOSS 1.00

```

```

                                b04f_call.cdi
INITIAL INDEX    LIBOR_1YR      2.2
!
DEFINE TRANCHE "SUBORD_1", "SUBORD_2", "SUBORD_3", "SUBORD_4", "RESID", "1A1",
"2A1", "GMC_2A1", "2A2", "GMC_2A2", "2A3", "GMC_2A3", "2A4", "GMC_2A4", "2A5",
"GMC_2A5", "2A6", "GMC_2A6", "2A7", "3A1", "4A1", "SUBS", "2WAC-IO"
!
!
Tranche "SUBORD_1" MODELING EXCHANGE
  Block 6766775.86 FLOAT GROUP 1 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate1 )
    0      999
!
Tranche "SUBORD_2" MODELING EXCHANGE
  Block 32192882.96 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0      999
!
Tranche "SUBORD_3" MODELING EXCHANGE
  Block 6351347.61 FLOAT GROUP 3 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate3 )
    0      999
!
Tranche "SUBORD_4" MODELING EXCHANGE
  Block 3217834.19 FLOAT GROUP 4 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate4 )
    0      999
!
Tranche "RESID" SEN_WAC
  Block 100.00 FLOAT GROUP 1 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate1 )
    0      999
!
Tranche "1A1" SEN_WAC
  Block 222600000.00 FLOAT GROUP 1 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate1 )
    0      999
!
Tranche "2A1" SEN_FLT_NO
  Block 135544000.00 GROUP 2 FREQ M FLOAT RESET M _
    DAYCOUNT 30360 BUSINESS_DAY NONE _
    Delay 0 Dated 20040625 Next 20040725
    (IF CURDATE LE 20090525 THEN min(2.341,COLL_NETRATE(2)) ELSE COLL_NETRATE(2))
    0      999
!
Tranche "GMC_2A1" EXCHANGE SEN_GMC_FLT
  Block 135544000.00 GROUP 2 FREQ M FLOAT RESET M _
    DAYCOUNT 30360 BUSINESS_DAY NONE _
    Delay 0 Dated 20040625 Next 20040725
    (IF CURDATE LE 20090525 THEN min(2.341,COLL_NETRATE(2)) ELSE COLL_NETRATE(2))
    0      999
!

```

b04f_call.cdi

```
Tranche "2A2" SEN_FLT_NO
Block 60180000.00 GROUP 2   FREQ M FLOAT RESET M _
    DAYCOUNT 30360 BUSINESS_DAY NONE _
    Delay 0 Dated 20040625 Next 20040725
    (IF CURDATE LE 20090525 then MIN(3.531,COLL_NETRATE(2)) ELSE COLL_NETRATE(2))
    0      999
!
Tranche "GMC_2A2" EXCHANGE SEN_GMC_FLT
Block 60180000.00 GROUP 2   FREQ M FLOAT RESET M _
    DAYCOUNT 30360 BUSINESS_DAY NONE _
    Delay 0 Dated 20040625 Next 20040725
    (IF CURDATE LE 20090525 then MIN(3.531,COLL_NETRATE(2)) ELSE COLL_NETRATE(2))
    0      999
!
Tranche "2A3" SEN_WAC_NO
Block 101379000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE   FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0      999
!
Tranche "GMC_2A3" EXCHANGE SEN_GMC_WAC
Block 101379000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE   FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0      999
!
Tranche "2A4" SEN_WAC_NO
Block 76411000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE   FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0      999
!
Tranche "GMC_2A4" EXCHANGE SEN_GMC_WAC
Block 76411000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE   FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0      999
!
Tranche "2A5" SEN_WAC_NO
Block 76789000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE   FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0      999
!
Tranche "GMC_2A5" EXCHANGE SEN_GMC_WAC
Block 76789000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE   FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0      999
!
Tranche "2A6" SEN_WAC_NO
Block 208788000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE   FREQ M _
    Delay 24 Dated 20040601 Next 20040725
    ( #NetRate2 )
    0      999
!
```

```

                                b04f_call.cdi
Tranche "GMC_2A6" EXCHANGE SEN_GMC_WAC
  Block 208788000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725 _
    ( #NetRate2 )
    0 999
!
Tranche "2A7" SEN_WAC
  Block 400000000.00 FLOAT GROUP 2 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725 _
    ( #NetRate2 )
    0 999
!
Tranche "3A1" SEN_WAC
  Block 208930000.00 FLOAT GROUP 3 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725 _
    ( #NetRate3 )
    0 999
!
Tranche "4A1" SEN_WAC
  Block 105839000.00 FLOAT GROUP 4 _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725 _
    ( #NetRate4 )
    0 999
!
Tranche "SUBS" JUN_WAC
  Block 48528840.62 FLOAT _
    DAYCOUNT 30360 BUSINESS_DAY NONE FREQ M _
    Delay 24 Dated 20040601 Next 20040725 _
    ( #COUPON_SUBS )
    0 999
!
Tranche "2WAC-IO" SEN_FLT_IO
  Block 195724000.00 GROUP 2 FREQ M FLOAT RESET M NOTIONAL WITH FORMULA BEGIN ( IF
CURMONTH LE 59 THEN BBAL("GMC_2A1#1","GMC_2A2#1") ELSE 0 ); _
                                                    END ( IF CURMONTH LT
59 THEN BBAL("GMC_2A1#1","GMC_2A2#1") ELSE 0 ); _
    DAYCOUNT 30360 BUSINESS_DAY NONE _
    Delay 24 Dated 20040601 Next 20040725 _
    (IF CURDATE LE 20090525 THEN
((COLL_NETRATE(2)-0)*BBAL("2A1","2A2")-COUPON("2A1")*BBAL("2A1") -
COUPON("2A2")*BBAL("2A2"))/BBAL("2A1","2A2") else 0)
    0 999
!
!
DEFINE PSEUDO_TRANCHE COLLAT _
  Delay 24 Dated 20040601 Next 20040725 settle 20040628
DEFINE PSEUDO_TRANCHE COLLAT GROUP 1 _
  Delay 24 Dated 20040601 Next 20040725 settle 20040628
DEFINE PSEUDO_TRANCHE COLLAT GROUP 2 _
  Delay 24 Dated 20040601 Next 20040725 settle 20040628
DEFINE PSEUDO_TRANCHE COLLAT GROUP 3 _
  Delay 24 Dated 20040601 Next 20040725 settle 20040628
DEFINE PSEUDO_TRANCHE COLLAT GROUP 4 _
  Delay 24 Dated 20040601 Next 20040725 settle 20040628
!
CREDIT_SUPPORT_BASIS GROUP_DEAL
!
ifdef #cmover_3.0g2 _
  DEFINE MACRO #BalanceRatio[1] = BALANCE_RATIO {#1}

```

b04f_call.cdi

```
!
ifndef #cmover_3.0g2 _
  DEFINE MACRO #BalanceRatio[1] =
  !
    CLASS "RESID"      NO_BUILD_TRANCHE _
                        = "RESID"
    CLASS "1A1"        NO_BUILD_TRANCHE _
                        = "1A1"
    CLASS "2WAC-IO"    NO_BUILD_TRANCHE _
                        = "2WAC-IO"
    CLASS "2A1"        NO_BUILD_TRANCHE _
                        = "2A1"
    CLASS "2A2"        NO_BUILD_TRANCHE _
                        = "2A2"
    CLASS "2A3"        NO_BUILD_TRANCHE _
                        = "2A3"
    CLASS "2A4"        NO_BUILD_TRANCHE _
                        = "2A4"
    CLASS "2A5"        NO_BUILD_TRANCHE _
                        = "2A5"
    CLASS "2A6"        NO_BUILD_TRANCHE _
                        = "2A6"
    CLASS "2A7"        NO_BUILD_TRANCHE _
                        = "2A7"
    CLASS "3A1"        NO_BUILD_TRANCHE _
                        = "3A1"
    CLASS "4A1"        NO_BUILD_TRANCHE _
                        = "4A1"
    CLASS "SUBORD_1"   DISTRIB_CLASS RULES _
                        = "SUBORD_1"
    CLASS "SUBORD_2"   DISTRIB_CLASS RULES _
                        = "SUBORD_2"
    CLASS "SUBORD_3"   DISTRIB_CLASS RULES _
                        = "SUBORD_3"
    CLASS "SUBORD_4"   DISTRIB_CLASS RULES _
                        = "SUBORD_4"
    CLASS "SEQ" DISTRIB_CLASS PRORATA WRITEDOWN_BAL PRORATA _
                        = "2A1" "2A2" "2A3" "2A4" "2A5" "2A6"
    CLASS "PT" _
                        = "2A7"
    CLASS "SUBS"       NO_BUILD_TRANCHE _
                        = "SUBS", _
    COMBINE_CLASSES {#BalanceRatio}{1.0} = "SUBORD_1" "SUBORD_2" "SUBORD_3"
    "SUBORD_4"
    CLASS "SNR_1"      WRITEDOWN_BAL PRORATA ALLOCATION WRITEDOWN_LIMIT BALANCE
    (#OrigCollBal1); _
                        = "RESID" "1A1", SUPPORT_CLASSES = "SUBS"
    CLASS "SNR_2"      WRITEDOWN_BAL PRORATA ALLOCATION WRITEDOWN_LIMIT BALANCE
    (#OrigCollBal2); _
                        = "SEQ" "PT" "2WAC-IO", SUPPORT_CLASSES = "SUBS"
    CLASS "SNR_3"      WRITEDOWN_BAL PRORATA ALLOCATION WRITEDOWN_LIMIT BALANCE
    (#OrigCollBal3); _
                        = "3A1", SUPPORT_CLASSES = "SUBS"
    CLASS "SNR_4"      WRITEDOWN_BAL PRORATA ALLOCATION WRITEDOWN_LIMIT BALANCE
    (#OrigCollBal4); _
                        = "4A1", SUPPORT_CLASSES = "SUBS"
  !
  !
  CLASS "GRP1" _
    DISTRIB_CLASS RULES _
    = "SNR_1" "SUBORD_1"
  CLASS "GRP2" _
    DISTRIB_CLASS RULES _
```

```

                                b04f_call.cdi
                                = "SNR_2" "SUBORD_2"
CLASS "GRP3" -
    DISTRIB_CLASS RULES
    = "SNR_3" "SUBORD_3"
CLASS "GRP4" -
    DISTRIB_CLASS RULES
    = "SNR_4" "SUBORD_4"
!
CLASS "ROOT" ROOT_LIST = "GRP1" "GRP2" "GRP3" "GRP4"
!
GROUP 0      ROOT      = 1 2 3 4
!

CLASS "GMC_2A1" PSEUDO NO_BUILD_TRANCHE = "GMC_2A1#1"
CLASS "GMC_2A2" PSEUDO NO_BUILD_TRANCHE = "GMC_2A2#1"
CLASS "GMC_2A3" PSEUDO NO_BUILD_TRANCHE = "GMC_2A3#1"
CLASS "GMC_2A4" PSEUDO NO_BUILD_TRANCHE = "GMC_2A4#1"
CLASS "GMC_2A5" PSEUDO NO_BUILD_TRANCHE = "GMC_2A5#1"
CLASS "GMC_2A6" PSEUDO NO_BUILD_TRANCHE = "GMC_2A6#1"
CLASS "GMC_2A7" PSEUDO NO_BUILD_TRANCHE = "2A7"
!
CLASS "GMC_SEQ" PSEUDO = "GMC_2A1" "GMC_2A2" "GMC_2A3" "GMC_2A4" "GMC_2A5"
"GMC_2A6"
CLASS "GMC_PT" PSEUDO = "GMC_2A7"
CLASS "GMC_SNR_2" PSEUDO = "GMC_SEQ" "PT" "2WAC-IO"
!
!
DEFINE PSEUDO_TRANCHE CLASS "SNR_1" Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SUBORD_1" Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SNR_2" Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SUBORD_2" Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SNR_3" Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SUBORD_3" Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SNR_4" Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SUBORD_4" Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "SEQ" Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "PT" Delay 24 Dated 20040601 Next 20040725
DAYCOUNT 30360 BUSINESS_DAY NONE
!
DEFINE PSEUDO_TRANCHE CLASS "GMC_SEQ" Delay 24 Dated 20040601 Next
20040725 DAYCOUNT 30360 BUSINESS_DAY NONE
DEFINE PSEUDO_TRANCHE CLASS "GMC_PT" Delay 24 Dated 20040601 Next
20040725 DAYCOUNT 30360 BUSINESS_DAY NONE
DEFINE PSEUDO_TRANCHE CLASS "GMC_SNR_2" Delay 24 Dated 20040601 Next
20040725 DAYCOUNT 30360 BUSINESS_DAY NONE

```

b04f_call.cdi

```

!
CROSSOVER when 0
!
DEFINE DYNAMIC STICKY #OrigSubBal1 = #OrigCollBal1 - ORIG_BBAL("SNR_1")
DEFINE DYNAMIC STICKY #SubBal1 = MAX(0, COLL_PREV_BAL(1) - BBAL("SNR_1"))
DEFINE DYNAMIC STICKY #OrigSubBal2 = #OrigCollBal2 - ORIG_BBAL("SNR_2")
DEFINE DYNAMIC STICKY #SubBal2 = MAX(0, COLL_PREV_BAL(2) - BBAL("SNR_2"))
DEFINE DYNAMIC STICKY #OrigSubBal3 = #OrigCollBal3 - ORIG_BBAL("SNR_3")
DEFINE DYNAMIC STICKY #SubBal3 = MAX(0, COLL_PREV_BAL(3) - BBAL("SNR_3"))
DEFINE DYNAMIC STICKY #OrigSubBal4 = #OrigCollBal4 - ORIG_BBAL("SNR_4")
DEFINE DYNAMIC STICKY #SubBal4 = MAX(0, COLL_PREV_BAL(4) - BBAL("SNR_4"))
DEFINE DYNAMIC STICKY #ReduceTestAagg = LOOKUP_TBL( "STEP", CURMONTH ,
"SI_LOSSAagg", "MONTH", "SHIFTR" )
DEFINE DYNAMIC STICKY #Sub2TimesLossPctAgg = LOOKUP_TBL( "STEP", CURMONTH ,
"SI_2TIMESLOSSAagg", "MONTH", "PCT" )
!

```

```

!
TRIGGER "CumLossAgg" _
    ORIG_TESTVAL 0.00% _
    TESTVAL (DELINQ_LOSS_ACCUM/ ORIG_BBAL("SUBS")); _
    TARGETVAL ( #ReduceTestAagg ); _
    ORIG_TARGETVAL NO_CHECK 20% _
    EFFECTIVE_WHEN ( IF CURMONTH GT 84 THEN TRIG_EFFECTIVE_YES ELSE
TRIG_EFFECTIVE_ALWAYSPASS ); _
    TRIGVAL LODIFF
!

```

```

!
TRIGGER "ShiftTriggerAgg" _
    FULL_NAME "Shifting Interest Loss Trigger" _
    DEFINITION "A Shifting Interest Loss Trigger will _

```

exist after month 84 if ;_

 cumulative losses as a percentage of the original subordinate _

bond balance are greater than the percentage in the following table; _

Month <=	%;	_
36	20%;	_
96	30%;	_
108	35%;	_
120	40%;	_
132	45%;	_
144	50% _	

The above calculations will be done based on all crossed mortgage loan groups." _

```

    EFFECTIVE_WHEN ( IF CURMONTH GT 84 THEN TRIG_EFFECTIVE_YES ELSE
TRIG_EFFECTIVE_ALWAYSPASS ); _
    IMPACT "Unscheduled principal that was intended to be paid to the _
subordinate bonds will now be directed to the senior bonds." _
    TRIGVAL FORMULA (TRIGGER("CumLossAgg","TRIGVAL"));
!

```

```

!
TRIGGER "TwoTimesCumLossAgg" _
    ORIG_TESTVAL 0.00% _
    TESTVAL (DELINQ_LOSS_ACCUM/(#OrigSubBal1 + #OrigSubBal2 +
#OrigSubBal3 + #OrigSubBal4)); _
    TARGETVAL ( #Sub2TimesLossPctAgg ); _
    ORIG_TARGETVAL NO_CHECK 20% _
    TRIGVAL LODIFF
!

```

```

!
TRIGGER "TwoTimesTriggerAgg" _
    FULL_NAME "Sub Two Times Loss Trigger" _
    DEFINITION "A Sub Two Times Loss Trigger will _

```

exist if ;_

 cumulative losses as a percentage of the original subordinate _

bond balance are greater than the percentage in the following table; _

Month <=	%;	_
36	20%;	_
96	30%;	_

```

                                b04f_call.cdi
                                108          35%; _
                                120          40%; _
                                132          45%; _
                                144          50%; _
IMPACT      "Unscheduled principal that was intended to be paid to the _
subordinate bonds will be directed to the senior bonds earlier than normal." _
TRIGVAL FORMULA (TRIGGER("TwoTimesCumLossAgg","TRIGVAL"));
!
!
DEFINE DYNAMIC STICKY #COUPON_SUBS = OPTIMAL_INTPMT("SUBORD_1", "SUBORD_2",
"SUBORD_3", "SUBORD_4") / BBAL("SUBORD_1", "SUBORD_2", "SUBORD_3", "SUBORD_4") *
36000 / NDAYS_ACCRUE_INT("SUBS#1")
!
OPTIONAL REDEMPTION:      "Cleanup" _
                           COLL_FRAC 10% _
                           PRICE_P ( COLL_BAL ); _
                           DISTR_P RULES "OPTR_DEAL"
!
!
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
!
INTEREST_SHORTFALL GROUP 3 FULL_PREPAY      Compensate Pro_rata _
                           PARTIAL_PREPAY Compensate Pro_rata _
                           LOSS              NO_Compensate SUBORDINATED ACCUM
!
INTEREST_SHORTFALL GROUP 4 FULL_PREPAY      Compensate Pro_rata _
                           PARTIAL_PREPAY Compensate Pro_rata _
                           LOSS              NO_Compensate SUBORDINATED ACCUM
!
Tranche      Cusip      Stated Maturity
SUBS          XXXXXXXXX  20340625
3A1           XXXXXXXXX  20340625
4A1           XXXXXXXXX  20340625
!
!
CMO Block Payment Rules
-----
!
calculate :  #SubsNotGoneAgg = BBAL("SUBS") GT 0.01
!
calculate :  #OrigSenPct1 = ORIG_BBAL("SNR_1") / #OrigCollBal1
!
calculate :  #SenPct1 = _
            IF #SubsNotGoneAgg _
            THEN MIN(1, BBAL("SNR_1") / COLL_PREV_BAL(1)) _
            ELSE 1
!
calculate :  #OrigSenPct2 = ORIG_BBAL("SNR_2") / #OrigCollBal2
!
calculate :  #SenPct2 = _
            IF #SubsNotGoneAgg _
            THEN MIN(1, BBAL("SNR_2") / COLL_PREV_BAL(2)) _
            ELSE 1
!
calculate :  #OrigSenPct3 = ORIG_BBAL("SNR_3") / #OrigCollBal3
!
calculate :  #SenPct3 = _

```

```

                                b04f_call.cdi
                                IF #SubsNotGoneAgg _
                                THEN MIN(1, BBAL("SNR_3") / COLL_PREV_BAL(3)) _
                                ELSE 1
!
! calculate : #OrigSenPct4 = ORIG_BBAL("SNR_4") / #OrigCollBal4
!
! calculate : #SenPct4 = _
!               IF #SubsNotGoneAgg _
!               THEN MIN(1, BBAL("SNR_4") / COLL_PREV_BAL(4)) _
!               ELSE 1
!
! calculate : #OrigSenPctAgg = ORIG_BBAL("SNR_1","SNR_2","SNR_3","SNR_4") /
! (#OrigCollBal1 + #OrigCollBal2 + #OrigCollBal3 + #OrigCollBal4)
! calculate : #SenPctAgg = MIN(1, BBAL("SNR_1","SNR_2","SNR_3","SNR_4") /
! (COLL_PREV_BAL))
!
! calculate : #ShiftTestAgg = NOT TRIGGER("ShiftTriggerAgg")
!
! calculate : #Sub2TimesTriggerAgg = NOT TRIGGER("TwoTimesTriggerAgg")
! calculate : #Sub2TimesTestAgg = BBAL("SUBORD_1", "SUBORD_2", "SUBORD_3",
! "SUBORD_4") / BBAL("SNR_1", "SUBORD_1", "SNR_2", "SUBORD_2", "SNR_3", "SUBORD_3",
! "SNR_4", "SUBORD_4") _
!               GE 2 * ORIG_BBAL("SUBORD_1", "SUBORD_2",
! "SUBORD_3", "SUBORD_4")/ORIG_BBAL("SNR_1", "SUBORD_1", "SNR_2", "SUBORD_2", "SNR_3",
! "SUBORD_3", "SNR_4", "SUBORD_4") AND _
!               #Sub2TimesTriggerAgg
!
! calculate : #SenPctFailAgg = (#SenPctAgg > #OrigSenPctAgg)
!
! calculate : #SenPrep1 = _
!               IF #SenPctFailAgg _
!               THEN 1 _
!               ELSE #SenPct1 + SHIFT%(1) * (1-#SenPct1), _
! Reduce_SHIFT%_when GROUP 1 STICKY_PASS FAILVAL_PRIOREND _
! (#ShiftTestAgg)
!
! calculate : #SenPrep2 = _
!               IF #SenPctFailAgg _
!               THEN 1 _
!               ELSE #SenPct2 + SHIFT%(2) * (1-#SenPct2), _
! Reduce_SHIFT%_when GROUP 2 STICKY_PASS FAILVAL_PRIOREND _
! (#ShiftTestAgg)
!
! calculate : #SenPrep3 = _
!               IF #SenPctFailAgg _
!               THEN 1 _
!               ELSE #SenPct3 + SHIFT%(3) * (1-#SenPct3), _
! Reduce_SHIFT%_when GROUP 3 STICKY_PASS FAILVAL_PRIOREND _
! (#ShiftTestAgg)
!
! calculate : #SenPrep4 = _
!               IF #SenPctFailAgg _
!               THEN 1 _
!               ELSE #SenPct4 + SHIFT%(4) * (1-#SenPct4), _
! Reduce_SHIFT%_when GROUP 4 STICKY_PASS FAILVAL_PRIOREND _
! (#ShiftTestAgg)
!
!
! calculate : #SenPrep1 = _
!               IF #SenPctFailAgg _
!               THEN 1 _
!               ELSE IF #Sub2TimesTestAgg _

```

```

                                b04f_call.cdi
                                THEN IF CURMONTH LE 36 _
                                    THEN #SenPct1 + (50% * (1-#SenPct1)) _
                                    ELSE #SenPct1 _
                                ELSE #SenPrep1
!
calculate : #SenPrep2 = _
            IF #SenPctFailAgg _
            THEN 1 _
            ELSE IF #Sub2TimesTestAgg _
                THEN IF CURMONTH LE 36 _
                    THEN #SenPct2 + (50% * (1-#SenPct2)) _
                    ELSE #SenPct2 _
                ELSE #SenPrep2
!
calculate : #SenPrep3 = _
            IF #SenPctFailAgg _
            THEN 1 _
            ELSE IF #Sub2TimesTestAgg _
                THEN IF CURMONTH LE 36 _
                    THEN #SenPct3 + (50% * (1-#SenPct3)) _
                    ELSE #SenPct3 _
                ELSE #SenPrep3
!
calculate : #SenPrep4 = _
            IF #SenPctFailAgg _
            THEN 1 _
            ELSE IF #Sub2TimesTestAgg _
                THEN IF CURMONTH LE 36 _
                    THEN #SenPct4 + (50% * (1-#SenPct4)) _
                    ELSE #SenPct4 _
                ELSE #SenPrep4
!
calculate : #SENRECOV1 = #SenPrep1 * DELINQ_RECOVER(1)
!
calculate : #SENRECOV2 = #SenPrep2 * DELINQ_RECOVER(2)
!
calculate : #SENRECOV3 = #SenPrep3 * DELINQ_RECOVER(3)
!
calculate : #SENRECOV4 = #SenPrep4 * DELINQ_RECOVER(4)
!
calculate: "SNR_1" _
NO_CHECK SCHEDULED _      GROUP 1 FRACTION LIMIT #SCH11 = #SenPct1 , _
NO_CHECK PREPAY          GROUP 1 FRACTION LIMIT #PRP11 = #SenPrep1 , _
NO_CHECK RECOVER         GROUP 1 AMOUNT   LIMIT #REC11 = #SENRECOV1
!
calculate: "SNR_2" _
NO_CHECK SCHEDULED _      GROUP 2 FRACTION LIMIT #SCH22 = #SenPct2 , _
NO_CHECK PREPAY          GROUP 2 FRACTION LIMIT #PRP22 = #SenPrep2 , _
NO_CHECK RECOVER         GROUP 2 AMOUNT   LIMIT #REC22 = #SENRECOV2
!
calculate: "SNR_3" _
NO_CHECK SCHEDULED _      GROUP 3 FRACTION LIMIT #SCH33 = #SenPct3 , _
NO_CHECK PREPAY          GROUP 3 FRACTION LIMIT #PRP33 = #SenPrep3 , _
NO_CHECK RECOVER         GROUP 3 AMOUNT   LIMIT #REC33 = #SENRECOV3
!
calculate: "SNR_4" _
NO_CHECK SCHEDULED _      GROUP 4 FRACTION LIMIT #SCH44 = #SenPct4 , _
NO_CHECK PREPAY          GROUP 4 FRACTION LIMIT #PRP44 = #SenPrep4 , _
NO_CHECK RECOVER         GROUP 4 AMOUNT   LIMIT #REC44 = #SENRECOV4
!
calculate : #SenSchedAlloc1 = #SCH11 * COLL_P_SCHED(1)
calculate : #SenPrepayAlloc1 = #PRP11 * COLL_P_PREPAY(1)
calculate : #SenRecoverAlloc1 = #REC11

```

```

                                b04f_call.cdi
calculate : #SenSchedAlloc2    = #SCH22 * COLL_P_SCHED(2)
calculate : #SenPrepayAlloc2   = #PRP22 * COLL_P_PREPAY(2)
calculate : #SenRecoverAlloc2  = #REC22
calculate : #SenSchedAlloc3    = #SCH33 * COLL_P_SCHED(3)
calculate : #SenPrepayAlloc3   = #PRP33 * COLL_P_PREPAY(3)
calculate : #SenRecoverAlloc3  = #REC33
calculate : #SenSchedAlloc4    = #SCH44 * COLL_P_SCHED(4)
calculate : #SenPrepayAlloc4   = #PRP44 * COLL_P_PREPAY(4)
calculate : #SenRecoverAlloc4  = #REC44
!
!! ***** CROSS SENIORS DUE TO RAPID PREPAYS *****
!
calculate : #SubMultipleAgg = LOOKUP_TBL( "STEP", CURMONTH ,
"CROSSPREP_2TIMESAgg", "MONTH", "MULTIPLE" )
!
calculate : #CrossTestAgg = ((#SubBal1 + #SubBal2 + #SubBal3 + #SubBal4) /
(COLL_PREV_BAL(1) + COLL_PREV_BAL(2) + COLL_PREV_BAL(3) + COLL_PREV_BAL(4)) LT _
#SubMultipleAgg * (#OrigSubBal1 + #OrigSubBal2 + #OrigSubBal3 +
#OrigSubBal4) / (#OrigCollBal1 + #OrigCollBal2 + #OrigCollBal3 + #OrigCollBal4))
!
calculate : #CrossTest1 = _
BBAL("SNR_1") LT 0.01 AND #CrossTestAgg
!
calculate : #CrossTest2 = _
BBAL("SNR_2") LT 0.01 AND #CrossTestAgg
!
calculate : #CrossTest3 = _
BBAL("SNR_3") LT 0.01 AND #CrossTestAgg
!
calculate : #CrossTest4 = _
BBAL("SNR_4") LT 0.01 AND #CrossTestAgg
!
calculate: "SNR_1" _
SCHEDULED GROUP 2 AMOUNT LIMIT #CROSS2SCH1 = IF #CrossTest2 THEN
(COLL_P_SCHED(2) - #SenSchedAlloc2) * BBAL("SNR_1")/BBAL("SNR_1","SNR_3","SNR_4")
ELSE 0 ; _
PREPAY GROUP 2 AMOUNT LIMIT #CROSS2PRP1 = IF #CrossTest2 THEN
(COLL_P_PREPAY(2) - #SenPrepayAlloc2) * BBAL("SNR_1")/BBAL("SNR_1","SNR_3","SNR_4")
ELSE 0 ; _
RECOVER GROUP 2 AMOUNT LIMIT #CROSS2REC1 = IF #CrossTest2 THEN
(DELINQ_RECOVER(2) - #SenRecoverAlloc2) *
BBAL("SNR_1")/BBAL("SNR_1","SNR_3","SNR_4") ELSE 0 ; _
SCHEDULED GROUP 3 AMOUNT LIMIT #CROSS3SCH1 = IF #CrossTest3 THEN
(COLL_P_SCHED(3) - #SenSchedAlloc3) * BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_4")
ELSE 0 ; _
PREPAY GROUP 3 AMOUNT LIMIT #CROSS3PRP1 = IF #CrossTest3 THEN
(COLL_P_PREPAY(3) - #SenPrepayAlloc3) * BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_4")
ELSE 0 ; _
RECOVER GROUP 3 AMOUNT LIMIT #CROSS3REC1 = IF #CrossTest3 THEN
(DELINQ_RECOVER(3) - #SenRecoverAlloc3) *
BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_4") ELSE 0 ; _
SCHEDULED GROUP 4 AMOUNT LIMIT #CROSS4SCH1 = IF #CrossTest4 THEN
(COLL_P_SCHED(4) - #SenSchedAlloc4) * BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; _
PREPAY GROUP 4 AMOUNT LIMIT #CROSS4PRP1 = IF #CrossTest4 THEN
(COLL_P_PREPAY(4) - #SenPrepayAlloc4) * BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; _
RECOVER GROUP 4 AMOUNT LIMIT #CROSS4REC1 = IF #CrossTest4 THEN
(DELINQ_RECOVER(4) - #SenRecoverAlloc4) *
BBAL("SNR_1")/BBAL("SNR_1","SNR_2","SNR_3") ELSE 0
!
calculate: "SNR_2" _

```

```

                                b04f_call.cdi
SCHEDULED GROUP 1 AMOUNT LIMIT #CROSS1SCH2 = IF #CrossTest1 THEN
(COLL_P_SCHED(1) - #SenSchedAlloc1) * BBAL("SNR_2")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
PREPAY GROUP 1 AMOUNT LIMIT #CROSS1PRP2 = IF #CrossTest1 THEN
(COLL_P_PREPAY(1) - #SenPrepayAlloc1) * BBAL("SNR_2")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
RECOVER GROUP 1 AMOUNT LIMIT #CROSS1REC2 = IF #CrossTest1 THEN
(DELINQ_RECOVER(1) - #SenRecoverAlloc1) *
BBAL("SNR_2")/BBAL("SNR_2","SNR_3","SNR_4") ELSE 0 ; -
SCHEDULED GROUP 3 AMOUNT LIMIT #CROSS3SCH2 = IF #CrossTest3 THEN
(COLL_P_SCHED(3) - #SenSchedAlloc3) * BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_4")
ELSE 0 ; -
PREPAY GROUP 3 AMOUNT LIMIT #CROSS3PRP2 = IF #CrossTest3 THEN
(COLL_P_PREPAY(3) - #SenPrepayAlloc3) * BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_4")
ELSE 0 ; -
RECOVER GROUP 3 AMOUNT LIMIT #CROSS3REC2 = IF #CrossTest3 THEN
(DELINQ_RECOVER(3) - #SenRecoverAlloc3) *
BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_4") ELSE 0 ; -
SCHEDULED GROUP 4 AMOUNT LIMIT #CROSS4SCH2 = IF #CrossTest4 THEN
(COLL_P_SCHED(4) - #SenSchedAlloc4) * BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; -
PREPAY GROUP 4 AMOUNT LIMIT #CROSS4PRP2 = IF #CrossTest4 THEN
(COLL_P_PREPAY(4) - #SenPrepayAlloc4) * BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; -
RECOVER GROUP 4 AMOUNT LIMIT #CROSS4REC2 = IF #CrossTest4 THEN
(DELINQ_RECOVER(4) - #SenRecoverAlloc4) *
BBAL("SNR_2")/BBAL("SNR_1","SNR_2","SNR_3") ELSE 0
!
calculate: "SNR_3" -
SCHEDULED GROUP 1 AMOUNT LIMIT #CROSS1SCH3 = IF #CrossTest1 THEN
(COLL_P_SCHED(1) - #SenSchedAlloc1) * BBAL("SNR_3")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
PREPAY GROUP 1 AMOUNT LIMIT #CROSS1PRP3 = IF #CrossTest1 THEN
(COLL_P_PREPAY(1) - #SenPrepayAlloc1) * BBAL("SNR_3")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
RECOVER GROUP 1 AMOUNT LIMIT #CROSS1REC3 = IF #CrossTest1 THEN
(DELINQ_RECOVER(1) - #SenRecoverAlloc1) *
BBAL("SNR_3")/BBAL("SNR_2","SNR_3","SNR_4") ELSE 0 ; -
SCHEDULED GROUP 2 AMOUNT LIMIT #CROSS2SCH3 = IF #CrossTest2 THEN
(COLL_P_SCHED(2) - #SenSchedAlloc2) * BBAL("SNR_3")/BBAL("SNR_1","SNR_3","SNR_4")
ELSE 0 ; -
PREPAY GROUP 2 AMOUNT LIMIT #CROSS2PRP3 = IF #CrossTest2 THEN
(COLL_P_PREPAY(2) - #SenPrepayAlloc2) * BBAL("SNR_3")/BBAL("SNR_1","SNR_3","SNR_4")
ELSE 0 ; -
RECOVER GROUP 2 AMOUNT LIMIT #CROSS2REC3 = IF #CrossTest2 THEN
(DELINQ_RECOVER(2) - #SenRecoverAlloc2) *
BBAL("SNR_3")/BBAL("SNR_1","SNR_3","SNR_4") ELSE 0 ; -
SCHEDULED GROUP 4 AMOUNT LIMIT #CROSS4SCH3 = IF #CrossTest4 THEN
(COLL_P_SCHED(4) - #SenSchedAlloc4) * BBAL("SNR_3")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; -
PREPAY GROUP 4 AMOUNT LIMIT #CROSS4PRP3 = IF #CrossTest4 THEN
(COLL_P_PREPAY(4) - #SenPrepayAlloc4) * BBAL("SNR_3")/BBAL("SNR_1","SNR_2","SNR_3")
ELSE 0 ; -
RECOVER GROUP 4 AMOUNT LIMIT #CROSS4REC3 = IF #CrossTest4 THEN
(DELINQ_RECOVER(4) - #SenRecoverAlloc4) *
BBAL("SNR_3")/BBAL("SNR_1","SNR_2","SNR_3") ELSE 0
!
calculate: "SNR_4" -
SCHEDULED GROUP 1 AMOUNT LIMIT #CROSS1SCH4 = IF #CrossTest1 THEN
(COLL_P_SCHED(1) - #SenSchedAlloc1) * BBAL("SNR_4")/BBAL("SNR_2","SNR_3","SNR_4")
ELSE 0 ; -
PREPAY GROUP 1 AMOUNT LIMIT #CROSS1PRP4 = IF #CrossTest1 THEN
(COLL_P_PREPAY(1) - #SenPrepayAlloc1) * BBAL("SNR_4")/BBAL("SNR_2","SNR_3","SNR_4")

```

b04f_call.cdi

```

ELSE 0 ; -
    RECOVER      GROUP 1  AMOUNT LIMIT #CROSS1REC4 = IF #CrossTest1 THEN
(DELINQ_RECOVER(1) - #SenRecoverAlloc1) *
BBAL("SNR_4")/BBAL("SNR_2","SNR_3","SNR_4") ELSE 0 ; -
    SCHEDULED    GROUP 2  AMOUNT LIMIT #CROSS2SCH4 = IF #CrossTest2 THEN
(COLL_P_SCHED(2) - #SenSchedAlloc2) * BBAL("SNR_4")/BBAL("SNR_1","SNR_3","SNR_4")
    ELSE 0 ; -
    PREPAY       GROUP 2  AMOUNT LIMIT #CROSS2PRP4 = IF #CrossTest2 THEN
(COLL_P_PREPAY(2) - #SenPrepayAlloc2) * BBAL("SNR_4")/BBAL("SNR_1","SNR_3","SNR_4")
    ELSE 0 ; -
    RECOVER      GROUP 2  AMOUNT LIMIT #CROSS2REC4 = IF #CrossTest2 THEN
(DELINQ_RECOVER(2) - #SenRecoverAlloc2) *
BBAL("SNR_4")/BBAL("SNR_1","SNR_3","SNR_4") ELSE 0 ; -
    SCHEDULED    GROUP 3  AMOUNT LIMIT #CROSS3SCH4 = IF #CrossTest3 THEN
(COLL_P_SCHED(3) - #SenSchedAlloc3) * BBAL("SNR_4")/BBAL("SNR_1","SNR_2","SNR_4")
    ELSE 0 ; -
    PREPAY       GROUP 3  AMOUNT LIMIT #CROSS3PRP4 = IF #CrossTest3 THEN
(COLL_P_PREPAY(3) - #SenPrepayAlloc3) * BBAL("SNR_4")/BBAL("SNR_1","SNR_2","SNR_4")
    ELSE 0 ; -
    RECOVER      GROUP 3  AMOUNT LIMIT #CROSS3REC4 = IF #CrossTest3 THEN
(DELINQ_RECOVER(3) - #SenRecoverAlloc3) *
BBAL("SNR_4")/BBAL("SNR_1","SNR_2","SNR_4") ELSE 0
!
    calculate : #SenSchedAlloc1 = #SenSchedAlloc1 + #CROSS1SCH2 + #CROSS1SCH3 +
#CROSS1SCH4
    calculate : #SenPrepayAlloc1 = #SenPrepayAlloc1 + #CROSS1PRP2 + #CROSS1PRP3 +
#CROSS1PRP4
    calculate : #SenRecoverAlloc1 = #SenRecoverAlloc1 + #CROSS1REC2 + #CROSS1REC3 +
#CROSS1REC4
!
    calculate : #SenSchedAlloc2 = #SenSchedAlloc2 + #CROSS2SCH1 + #CROSS2SCH3 +
#CROSS2SCH4
    calculate : #SenPrepayAlloc2 = #SenPrepayAlloc2 + #CROSS2PRP1 + #CROSS2PRP3 +
#CROSS2PRP4
    calculate : #SenRecoverAlloc2 = #SenRecoverAlloc2 + #CROSS2REC1 + #CROSS2REC3 +
#CROSS2REC4
!
    calculate : #SenSchedAlloc3 = #SenSchedAlloc3 + #CROSS3SCH1 + #CROSS3SCH2 +
#CROSS3SCH4
    calculate : #SenPrepayAlloc3 = #SenPrepayAlloc3 + #CROSS3PRP1 + #CROSS3PRP2 +
#CROSS3PRP4
    calculate : #SenRecoverAlloc3 = #SenRecoverAlloc3 + #CROSS3REC1 + #CROSS3REC2 +
#CROSS3REC4
!
    calculate : #SenSchedAlloc4 = #SenSchedAlloc4 + #CROSS4SCH1 + #CROSS4SCH2 +
#CROSS4SCH3
    calculate : #SenPrepayAlloc4 = #SenPrepayAlloc4 + #CROSS4PRP1 + #CROSS4PRP2 +
#CROSS4PRP3
    calculate : #SenRecoverAlloc4 = #SenRecoverAlloc4 + #CROSS4REC1 + #CROSS4REC2 +
#CROSS4REC3
!
!! ***** END CROSS SENIORS DUE TO RAPID PREPAYS *****
!
    calculate : #SubSched1 = MAX( 0, COLL_P_SCHED(1) - #SenSchedAlloc1 )
    calculate : #SubPrepay1 = MAX( 0, COLL_P_PREPAY(1) - #SenPrepayAlloc1 )
    calculate : #SubRecov1 = MAX( 0, DELINQ_RECOVER(1) - #SenRecoverAlloc1 )
!
    calculate : #SubSched2 = MAX( 0, COLL_P_SCHED(2) - #SenSchedAlloc2 )
    calculate : #SubPrepay2 = MAX( 0, COLL_P_PREPAY(2) - #SenPrepayAlloc2 )
    calculate : #SubRecov2 = MAX( 0, DELINQ_RECOVER(2) - #SenRecoverAlloc2 )
!
    calculate : #SubSched3 = MAX( 0, COLL_P_SCHED(3) - #SenSchedAlloc3 )

```

```

                                b04f_call.cdi
calculate : #SubPrepay3 = MAX( 0, COLL_P_PREPAY(3) - #SenPrepayAlloc3 )
calculate : #SubRecov3 = MAX( 0, DELINQ_RECOVER(3) - #SenRecoverAlloc3 )
!
calculate : #SubSched4 = MAX( 0, COLL_P_SCHED(4) - #SenSchedAlloc4 )
calculate : #SubPrepay4 = MAX( 0, COLL_P_PREPAY(4) - #SenPrepayAlloc4 )
calculate : #SubRecov4 = MAX( 0, DELINQ_RECOVER(4) - #SenRecoverAlloc4 )
!
calculate: "SUBORD_1" _
NO_CHECK SCHEDULED      GROUP 1  AMOUNT          = #SubSched1 , _
NO_CHECK PREPAY         GROUP 1  AMOUNT          = #SubPrepay1 , _
NO_CHECK RECOVER        GROUP 1  AMOUNT          = #SubRecov1
!
calculate: "SUBORD_2" _
NO_CHECK SCHEDULED      GROUP 2  AMOUNT          = #SubSched2 , _
NO_CHECK PREPAY         GROUP 2  AMOUNT          = #SubPrepay2 , _
NO_CHECK RECOVER        GROUP 2  AMOUNT          = #SubRecov2
!
calculate: "SUBORD_3" _
NO_CHECK SCHEDULED      GROUP 3  AMOUNT          = #SubSched3 , _
NO_CHECK PREPAY         GROUP 3  AMOUNT          = #SubPrepay3 , _
NO_CHECK RECOVER        GROUP 3  AMOUNT          = #SubRecov3
!
calculate: "SUBORD_4" _
NO_CHECK SCHEDULED      GROUP 4  AMOUNT          = #SubSched4 , _
NO_CHECK PREPAY         GROUP 4  AMOUNT          = #SubPrepay4 , _
NO_CHECK RECOVER        GROUP 4  AMOUNT          = #SubRecov4
!
calculate : #SubSched      = #SubSched1 + #SubSched2 + #SubSched3 + #SubSched4
calculate : #SubPrepay     = #SubPrepay1 + #SubPrepay2 + #SubPrepay3 + #SubPrepay4
calculate : #SubRecov      = #SubRecov1 + #SubRecov2 + #SubRecov3 + #SubRecov4
!
calculate: "SUBS" _
NO_CHECK SCHEDULED      AMOUNT          = #SubSched , _
NO_CHECK PREPAY         AMOUNT          = #SubPrepay , _
NO_CHECK RECOVER        AMOUNT          = #SubRecov
!
calculate : #ExcessSub     = EXCESS_SUB_AMT
!
calculate : #ExcessSub     = EXCESS_SUB_AMT("SUBS")
!
calculate : #ExcessSub1    = #ExcessSub * BBAL("SNR_1") /
BBAL("SNR_1","SNR_2","SNR_3","SNR_4")
calculate : #ExcessSub2    = #ExcessSub * BBAL("SNR_2") /
BBAL("SNR_1","SNR_2","SNR_3","SNR_4")
calculate : #ExcessSub3    = #ExcessSub * BBAL("SNR_3") /
BBAL("SNR_1","SNR_2","SNR_3","SNR_4")
calculate : #ExcessSub4    = #ExcessSub * BBAL("SNR_4") /
BBAL("SNR_1","SNR_2","SNR_3","SNR_4")
!
calculate : "SNR_1" _
NO_CHECK EXCESS_SUB_AMT AMOUNT LIMIT #VS1 = #ExcessSub1
!
calculate : "SNR_2" _
NO_CHECK EXCESS_SUB_AMT AMOUNT LIMIT #VS2 = #ExcessSub2
!
calculate : "SNR_3" _
NO_CHECK EXCESS_SUB_AMT AMOUNT LIMIT #VS3 = #ExcessSub3
!
calculate : "SNR_4" _
NO_CHECK EXCESS_SUB_AMT AMOUNT LIMIT #VS4 = #ExcessSub4
!
calculate : #SubExcessSub1  = MIN(OPTIMAL_PRINCPMT("SUBORD_1"), #VS1)
calculate : #RemSubExcessSub1 = #VS1 - #SubExcessSub1

```

```

                                b04f_call1.cdi
calculate : #RemSubOptPrinc1 = OPTIMAL_PRINCPMT("SUBORD_1") - #SubExcessSub1
calculate : #SubExcessSub2 = MIN(OPTIMAL_PRINCPMT("SUBORD_2"), #VS2)
calculate : #RemSubExcessSub2 = #VS2 - #SubExcessSub2
calculate : #RemSubOptPrinc2 = OPTIMAL_PRINCPMT("SUBORD_2") - #SubExcessSub2
calculate : #SubExcessSub3 = MIN(OPTIMAL_PRINCPMT("SUBORD_3"), #VS3)
calculate : #RemSubExcessSub3 = #VS3 - #SubExcessSub3
calculate : #RemSubOptPrinc3 = OPTIMAL_PRINCPMT("SUBORD_3") - #SubExcessSub3
calculate : #SubExcessSub4 = MIN(OPTIMAL_PRINCPMT("SUBORD_4"), #VS4)
calculate : #RemSubExcessSub4 = #VS4 - #SubExcessSub4
calculate : #RemSubOptPrinc4 = OPTIMAL_PRINCPMT("SUBORD_4") - #SubExcessSub4
calculate : #RemSubExcessSub = #RemSubExcessSub1 + #RemSubExcessSub2 +
#RemSubExcessSub3 + #RemSubExcessSub4
calculate : #RemSubOptPrinc = #RemSubOptPrinc1 + #RemSubOptPrinc2 +
#RemSubOptPrinc3 + #RemSubOptPrinc4
!
calculate : #TotSubExcessSub1 = -1 * (#SubExcessSub1 + #RemSubExcessSub *
#RemSubOptPrinc1/#RemSubOptPrinc)
calculate : #TotSubExcessSub2 = -1 * (#SubExcessSub2 + #RemSubExcessSub *
#RemSubOptPrinc2/#RemSubOptPrinc)
calculate : #TotSubExcessSub3 = -1 * (#SubExcessSub3 + #RemSubExcessSub *
#RemSubOptPrinc3/#RemSubOptPrinc)
calculate : #TotSubExcessSub4 = -1 * (#SubExcessSub4 + #RemSubExcessSub *
#RemSubOptPrinc4/#RemSubOptPrinc)
!
calculate : "SUBORD_1" _
NO_CHECK NEG_OK CUSTOM AMOUNT = #TotSubExcessSub1
!
calculate : "SUBORD_2" _
NO_CHECK NEG_OK CUSTOM AMOUNT = #TotSubExcessSub2
!
calculate : "SUBORD_3" _
NO_CHECK NEG_OK CUSTOM AMOUNT = #TotSubExcessSub3
!
calculate : "SUBORD_4" _
NO_CHECK NEG_OK CUSTOM AMOUNT = #TotSubExcessSub4
!
-----
pay : CLASS INTEREST PRO_RATE ("SNR_1" )
pay : CLASS INTSHORT PRO_RATE ("SNR_1" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "SNR_1" )
-----
pay : CLASS INTEREST PRO_RATE ("SNR_2" )
pay : CLASS INTSHORT PRO_RATE ("SNR_2" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "SNR_2" )
-----
pay : CLASS INTEREST PRO_RATE ("SNR_3" )
pay : CLASS INTSHORT PRO_RATE ("SNR_3" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "SNR_3" )
-----
pay : CLASS INTEREST PRO_RATE ("SNR_4" )
pay : CLASS INTSHORT PRO_RATE ("SNR_4" )
pay : CLASS PRINCIPAL SEQUENTIAL ( "SNR_4" )
-----
pay : CLASS INTEREST PRO_RATE ( "SUBORD_1" )
pay : CLASS INTSHORT PRO_RATE ( "SUBORD_1" )
-----
pay : CLASS INTEREST PRO_RATE ( "SUBORD_2" )
pay : CLASS INTSHORT PRO_RATE ( "SUBORD_2" )
-----
pay : CLASS INTEREST PRO_RATE ( "SUBORD_3" )
pay : CLASS INTSHORT PRO_RATE ( "SUBORD_3" )
-----
pay : CLASS INTEREST PRO_RATE ( "SUBORD_4" )

```

```

                                b04f_call.cdi
pay : CLASS INTSHORT PRO_RATA ( "SUBORD_4" )
-----
from : CLASS ( "GRP1"; "GRP2"; "GRP3"; "GRP4" )
pay : CLASS INTEREST PRO_RATA ( "SUBORD_1"; "SUBORD_2"; "SUBORD_3";
"SUBORD_4" )
pay : CLASS INTSHORT PRO_RATA ( "SUBORD_1"; "SUBORD_2"; "SUBORD_3";
"SUBORD_4" )
-----
from : CLASS ( "GRP1" ; "GRP2" ; "GRP3" ; "GRP4" )
pay : CLASS ENTIRETY PRO_RATA ( "SNR_1" ; "SNR_2" ; "SNR_3" ; "SNR_4" )
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "SUBORD_1" )
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "SUBORD_2" )
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "SUBORD_3" )
-----
pay : CLASS PRINCIPAL SEQUENTIAL ( "SUBORD_4" )
-----
from : CLASS ( "GRP1"; "GRP2"; "GRP3"; "GRP4" )
pay : CLASS PRINCIPAL PRO_RATA ( "SUBORD_1"; "SUBORD_2"; "SUBORD_3";
"SUBORD_4" )
-----
from : CLASS ( "SNR_1" )
pay : CLASS INTEREST PRO_RATA ( "RESID"; "1A1" )
pay : CLASS INTSHORT PRO_RATA ( "RESID"; "1A1" )
-----
from : CLASS ( "SNR_1" )
pay : CLASS BALANCE SEQUENTIAL ( "RESID", "1A1" )
-----
!
-----
from : CLASS ( "RESID" )
pay : SEQUENTIAL ( "RESID#1" )
-----
from : CLASS ( "1A1" )
pay : SEQUENTIAL ( "1A1#1" )
-----
subject to : CEILING ( (BBAL("SUBORD_1#1")-BBAL("SUBORD_1")) )
pay : SEQUENTIAL ( "SUBORD_1#1" )
-----
!
from : CLASS ( "SNR_2" )
pay : CLASS INTEREST PRO_RATA ( "SEQ"; "PT"; "2WAC-IO" )
pay : CLASS INTSHORT PRO_RATA ( "SEQ"; "PT"; "2WAC-IO" )
-----
from : CLASS ( "SNR_2" )
pay : CLASS BALANCE PRO_RATA ( "SEQ" ; "PT" )
-----
!
from : CLASS ( "SEQ" )
pay : CLASS INTEREST PRO_RATA ( "2A1"; "2A2"; "2A3"; "2A4"; "2A5"; "2A6"
)
pay : CLASS INTSHORT PRO_RATA ( "2A1"; "2A2"; "2A3"; "2A4"; "2A5"; "2A6"
)
-----
from : CLASS ( "SEQ" )
pay : CLASS BALANCE SEQUENTIAL ( "2A1", "2A2", "2A3", "2A4", "2A5", "2A6"
)
-----
!
from : CLASS ( "PT" )
pay : CLASS INTEREST PRO_RATA ( "2A7" )

```

```

                                b04f_call.cdi
pay : CLASS INTSHORT PRO_RATA ( "2A7" )
-----
from : CLASS ( "PT" )
pay : CLASS BALANCE SEQUENTIAL ( "2A7" )
-----
!
-----
from : CLASS ( "2A1" )
pay : SEQUENTIAL ( "2A1#1" )
-----
from : CLASS ( "2A2" )
pay : SEQUENTIAL ( "2A2#1" )
-----
from : CLASS ( "2A3" )
pay : SEQUENTIAL ( "2A3#1" )
-----
from : CLASS ( "2A4" )
pay : SEQUENTIAL ( "2A4#1" )
-----
from : CLASS ( "2A5" )
pay : SEQUENTIAL ( "2A5#1" )
-----
from : CLASS ( "2A6" )
pay : SEQUENTIAL ( "2A6#1" )
-----
from : CLASS ( "2A7" )
pay : SEQUENTIAL ( "2A7#1" )
-----
subject to : CEILING ( (BBAL("SUBORD_2#1")-BBAL("SUBORD_2"))) )
pay : SEQUENTIAL ( "SUBORD_2#1" )
-----
!
from : CLASS ( "SNR_3" )
pay : CLASS INTEREST PRO_RATA ( "3A1" )
pay : CLASS INTSHORT PRO_RATA ( "3A1" )
-----
from : CLASS ( "SNR_3" )
pay : CLASS BALANCE SEQUENTIAL ( "3A1" )
-----
!
-----
from : CLASS ( "3A1" )
pay : SEQUENTIAL ( "3A1#1" )
-----
subject to : CEILING ( (BBAL("SUBORD_3#1")-BBAL("SUBORD_3"))) )
pay : SEQUENTIAL ( "SUBORD_3#1" )
-----
!
from : CLASS ( "SNR_4" )
pay : CLASS INTEREST PRO_RATA ( "4A1" )
pay : CLASS INTSHORT PRO_RATA ( "4A1" )
-----
from : CLASS ( "SNR_4" )
pay : CLASS BALANCE SEQUENTIAL ( "4A1" )
-----
!
-----
from : CLASS ( "4A1" )
pay : SEQUENTIAL ( "4A1#1" )
-----
subject to : CEILING ( (BBAL("SUBORD_4#1")-BBAL("SUBORD_4"))) )
pay : SEQUENTIAL ( "SUBORD_4#1" )
-----

```

b04f_call.cdi

calculate : #P_SUBS =
PRINCPMT("SUBORD_1#1","SUBORD_2#1","SUBORD_3#1","SUBORD_4#1")

from : CLASS ("SUBORD_1" ; "SUBORD_2" ; "SUBORD_3" ; "SUBORD_4")
pay : CLASS INTEREST PRO_RATA ("SUBS")
pay : CLASS INTSHORT PRO_RATA ("SUBS")

from : CLASS ("SUBORD_1" ; "SUBORD_2" ; "SUBORD_3" ; "SUBORD_4")
subject to : CEILING (#P_SUBS)
pay : CLASS BALANCE SEQUENTIAL ("SUBS")

from : CLASS ("SUBORD_1" ; "SUBORD_2" ; "SUBORD_3" ; "SUBORD_4")
pay : CLASS MORE_INTEREST SEQUENTIAL ("GRP1")

!

from : CLASS ("SUBS")
pay : SEQUENTIAL ("SUBS#1")

!

----- PAYDOWN SUBORD TRANCHES
calculate : #PrincReduce = BBAL("SUBS#1") - BBAL("SUBORD_1", "SUBORD_2",
"SUBORD_3", "SUBORD_4")
calculate : #SubPrinc1 = BBAL("SUBORD_1#1") - BBAL("SUBORD_1")
calculate : #SubPrinc2 = BBAL("SUBORD_2#1") - BBAL("SUBORD_2")
calculate : #SubPrinc3 = BBAL("SUBORD_3#1") - BBAL("SUBORD_3")
calculate : #SubPrinc4 = BBAL("SUBORD_4#1") - BBAL("SUBORD_4")
calculate : #SubPrincAgg = #SubPrinc1 + #SubPrinc2 + #SubPrinc3 + #SubPrinc4
calculate : #PrincReduce1 = #PrincReduce * #SubPrinc1 / #SubPrincAgg
calculate : #PrincReduce2 = #PrincReduce * #SubPrinc2 / #SubPrincAgg
calculate : #PrincReduce3 = #PrincReduce * #SubPrinc3 / #SubPrincAgg
calculate : #PrincReduce4 = #PrincReduce * #SubPrinc4 / #SubPrincAgg

subject to : CEILING ((#SubPrinc1 - #PrincReduce1))
pay : SEQUENTIAL ("SUBORD_1#1")

subject to : CEILING ((#SubPrinc2 - #PrincReduce2))
pay : SEQUENTIAL ("SUBORD_2#1")

subject to : CEILING ((#SubPrinc3 - #PrincReduce3))
pay : SEQUENTIAL ("SUBORD_3#1")

subject to : CEILING ((#SubPrinc4 - #PrincReduce4))
pay : SEQUENTIAL ("SUBORD_4#1")

!

----- MANUAL WRITEDOWNS
calculate : #ReduceSubord1 = MAX(0, MIN(BBAL("SUBORD_1#1"), BBAL("RESID#1",
"1A1#1", "SUBORD_1#1") - COLL_BAL(1)))
calculate : #ReduceSubord2 = MAX(0, MIN(BBAL("SUBORD_2#1"), BBAL("2A1#1",
"2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1") - COLL_BAL(2)))
calculate : #ReduceSubord3 = MAX(0, MIN(BBAL("SUBORD_3#1"), BBAL("3A1#1",
"SUBORD_3#1") - COLL_BAL(3)))
calculate : #ReduceSubord4 = MAX(0, MIN(BBAL("SUBORD_4#1"), BBAL("4A1#1",
"SUBORD_4#1") - COLL_BAL(4)))
calculate : #TotReduceSubord = #ReduceSubord1 + #ReduceSubord2 + #ReduceSubord3
+ #ReduceSubord4

when : IS_THERE ("SUBS#1")
pay : DECREMENT (BALANCE "SUBORD_1#1", BY #ReduceSubord1)

calculate : #IncreaseSubord1 = MAX(0, MIN(#TotReduceSubord, COLL_BAL(1) -

```

                                b04f_call.cdi
BBAL( "RESID#1", "1A1#1", "SUBORD_1#1" )))
-----
    when :  IS_THERE ( "SUBS#1" )
    pay :  INCREMENT ( BALANCE "SUBORD_1#1", BY #IncreaseSubord1 )
-----
    when :  IS_THERE ( "SUBS#1" )
    pay :  DECREMENT ( BALANCE "SUBORD_2#1", BY #ReduceSubord2 )
-----
    calculate :  #IncreaseSubord2 = MAX(0, MIN( #TotReduceSubord, COLL_BAL(2) -
BBAL( "2A1#1", "2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1"
)))
-----
    when :  IS_THERE ( "SUBS#1" )
    pay :  INCREMENT ( BALANCE "SUBORD_2#1", BY #IncreaseSubord2 )
-----
    when :  IS_THERE ( "SUBS#1" )
    pay :  DECREMENT ( BALANCE "SUBORD_3#1", BY #ReduceSubord3 )
-----
    calculate :  #IncreaseSubord3 = MAX(0, MIN( #TotReduceSubord, COLL_BAL(3) -
BBAL( "3A1#1", "SUBORD_3#1" )))
-----
    when :  IS_THERE ( "SUBS#1" )
    pay :  INCREMENT ( BALANCE "SUBORD_3#1", BY #IncreaseSubord3 )
-----
    when :  IS_THERE ( "SUBS#1" )
    pay :  DECREMENT ( BALANCE "SUBORD_4#1", BY #ReduceSubord4 )
-----
    calculate :  #IncreaseSubord4 = MAX(0, MIN( #TotReduceSubord, COLL_BAL(4) -
BBAL( "4A1#1", "SUBORD_4#1" )))
-----
    when :  IS_THERE ( "SUBS#1" )
    pay :  INCREMENT ( BALANCE "SUBORD_4#1", BY #IncreaseSubord4 )
-----
    calculate :  #MoreReduceSubord1 = MAX(0, MIN( BBAL("SUBORD_1#1"),
BBAL("SUBORD_1#1") / BBAL("SUBORD_1#1", "SUBORD_2#1", "SUBORD_3#1", "SUBORD_4#1") *
( BBAL("RESID#1", "1A1#1", "SUBORD_1#1", "2A1#1", "2A2#1", "2A3#1", "2A4#1",
"2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1", "3A1#1", "SUBORD_3#1", "4A1#1",
"SUBORD_4#1") - COLL_BAL)))
    calculate :  #MoreReduceSubord2 = MAX(0, MIN( BBAL("SUBORD_2#1"),
BBAL("SUBORD_2#1") / BBAL("SUBORD_1#1", "SUBORD_2#1", "SUBORD_3#1", "SUBORD_4#1") *
( BBAL("RESID#1", "1A1#1", "SUBORD_1#1", "2A1#1", "2A2#1", "2A3#1", "2A4#1",
"2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1", "3A1#1", "SUBORD_3#1", "4A1#1",
"SUBORD_4#1") - COLL_BAL)))
    calculate :  #MoreReduceSubord3 = MAX(0, MIN( BBAL("SUBORD_3#1"),
BBAL("SUBORD_3#1") / BBAL("SUBORD_1#1", "SUBORD_2#1", "SUBORD_3#1", "SUBORD_4#1") *
( BBAL("RESID#1", "1A1#1", "SUBORD_1#1", "2A1#1", "2A2#1", "2A3#1", "2A4#1",
"2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1", "3A1#1", "SUBORD_3#1", "4A1#1",
"SUBORD_4#1") - COLL_BAL)))
    calculate :  #MoreReduceSubord4 = MAX(0, MIN( BBAL("SUBORD_4#1"),
BBAL("SUBORD_4#1") / BBAL("SUBORD_1#1", "SUBORD_2#1", "SUBORD_3#1", "SUBORD_4#1") *
( BBAL("RESID#1", "1A1#1", "SUBORD_1#1", "2A1#1", "2A2#1", "2A3#1", "2A4#1",
"2A5#1", "2A6#1", "2A7#1", "SUBORD_2#1", "3A1#1", "SUBORD_3#1", "4A1#1",
"SUBORD_4#1") - COLL_BAL)))
-----
    when :  IS_TRUE ( (BBAL("SUBS#1") GT 0.01 ) AND ( ( BBAL("SUBORD_2#1") LT
0.01 ) OR ( BBAL("SUBORD_3#1") LT 0.01 ) OR ( BBAL("SUBORD_4#1") LT 0.01 ) ) )
    pay :  DECREMENT ( BALANCE "SUBORD_1#1", BY #MoreReduceSubord1 )
-----
    when :  IS_TRUE ( (BBAL("SUBS#1") GT 0.01 ) AND ( ( BBAL("SUBORD_1#1") LT
0.01 ) OR ( BBAL("SUBORD_3#1") LT 0.01 ) OR ( BBAL("SUBORD_4#1") LT 0.01 ) ) )
    pay :  DECREMENT ( BALANCE "SUBORD_2#1", BY #MoreReduceSubord2 )
-----
    when :  IS_TRUE ( (BBAL("SUBS#1") GT 0.01 ) AND ( ( BBAL("SUBORD_1#1") LT

```

```

                                b04f_call.cdi
0.01 ) OR ( BBAL("SUBORD_2#1") LT 0.01 ) OR ( BBAL("SUBORD_4#1") LT 0.01 ) ))
    pay : DECREMENT ( BALANCE "SUBORD_3#1", BY #MoreReduceSubord3 )
-----
    when : IS_TRUE ( (BBAL("SUBS#1") GT 0.01 ) AND ( ( BBAL("SUBORD_1#1") LT
0.01 ) OR ( BBAL("SUBORD_2#1") LT 0.01 ) OR ( BBAL("SUBORD_3#1") LT 0.01 ) ))
    pay : DECREMENT ( BALANCE "SUBORD_4#1", BY #MoreReduceSubord4 )
-----
    calculate : #WriteDown = MAX(0, BBAL( "RESID#1", "1A1#1", "SUBS#1", "2A1#1",
"2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1", "3A1#1", "4A1#1" ) -
COLL_BAL(1,2,3,4))
-----
    from : SUBACCOUNT ( #WriteDown )
    pay : WRITEDOWN PRO_RATA ( "SUBS#1" )
-----
    calculate : #SenWriteDown1 = MAX(0, BBAL( "RESID#1", "1A1#1" ) / BBAL(
"RESID#1", "1A1#1", "2A1#1", "2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1",
"3A1#1", "4A1#1" ) * #WriteDown)
-----
    calculate : #SenWriteDown2 = MAX(0, BBAL( "2A1#1", "2A2#1", "2A3#1", "2A4#1",
"2A5#1", "2A6#1", "2A7#1" ) / BBAL( "RESID#1", "1A1#1", "2A1#1", "2A2#1", "2A3#1",
"2A4#1", "2A5#1", "2A6#1", "2A7#1", "3A1#1", "4A1#1" ) * #WriteDown)
-----
    calculate : #SenWriteDown3 = MAX(0, BBAL( "3A1#1" ) / BBAL( "RESID#1", "1A1#1",
"2A1#1", "2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1", "3A1#1", "4A1#1" ) *
#WriteDown)
-----
    calculate : #SenWriteDown4 = MAX(0, BBAL( "4A1#1" ) / BBAL( "RESID#1", "1A1#1",
"2A1#1", "2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1", "2A7#1", "3A1#1", "4A1#1" ) *
#WriteDown)
-----
    from : SUBACCOUNT ( #WriteDown, #SenWriteDown1 )
    pay : WRITEDOWN PRO_RATA ( "RESID#1"; "1A1#1" )
-----
    from : SUBACCOUNT ( #WriteDown, #SenWriteDown2 )
    pay : WRITEDOWN PRO_RATA ( "2A1#1"; "2A2#1"; "2A3#1"; "2A4#1"; "2A5#1";
"2A6#1"; "2A7#1" )
-----
    from : SUBACCOUNT ( #WriteDown, #SenWriteDown3 )
    pay : WRITEDOWN SEQUENTIAL ( "3A1#1" )
-----
    from : SUBACCOUNT ( #WriteDown, #SenWriteDown4 )
    pay : WRITEDOWN SEQUENTIAL ( "4A1#1" )
-----
!
    calculate : #BondBal1 = BBAL("RESID#1","1A1#1","SUBORD_1#1")
-----
    calculate : #BondBal2 =
BBAL("2A1#1","2A2#1","2A3#1","2A4#1","2A5#1","2A6#1","2A7#1","SUBORD_2#1")
-----
    calculate : #BondBal3 = BBAL("3A1#1","SUBORD_3#1")
-----
    calculate : #BondBal4 = BBAL("4A1#1","SUBORD_4#1")
-----
    calculate : #BondBal = #BondBal1 + #BondBal2 + #BondBal3 + #BondBal4
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (INTPMT("2A1#1")) )
    pay : INTEREST SEQUENTIAL ("GMC_2A1#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A1#1")) )
    pay : SEQUENTIAL ("GMC_2A1#1")
-----

```

```

                                b04f_call.cdi
calculate : #GMCWD = BBAL("GMC_2A1#1") - BBAL("2A1#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A1#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A1#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A1#1")) )
    pay : SEQUENTIAL ("GMC_2A1#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A1#1") * COUPON("2A1#1") / 1200 * 0 / 30) )
    pay : AS_INTEREST ("GMC_2A1#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (INTPMT("2A2#1")) )
    pay : INTEREST SEQUENTIAL ("GMC_2A2#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A2#1")) )
    pay : SEQUENTIAL ("GMC_2A2#1")
-----
calculate : #GMCWD = BBAL("GMC_2A2#1") - BBAL("2A2#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A2#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A2#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A2#1")) )
    pay : SEQUENTIAL ("GMC_2A2#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A2#1") * COUPON("2A2#1") / 1200 * 0 / 30) )
    pay : AS_INTEREST ("GMC_2A2#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (INTPMT("2A3#1")) )
    pay : INTEREST SEQUENTIAL ("GMC_2A3#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A3#1")) )
    pay : SEQUENTIAL ("GMC_2A3#1")
-----
calculate : #GMCWD = BBAL("GMC_2A3#1") - BBAL("2A3#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )

```

```

                                b04f_call.cdi
pay : WRITEDOWN SEQUENTIAL ("GMC_2A3#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A3#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A3#1")) )
    pay : SEQUENTIAL ("GMC_2A3#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A3#1") * COUPON("2A3#1") / 1200 * 24 / 30) )
    pay : AS_INTEREST ("GMC_2A3#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (INTPMT("2A4#1")) )
    pay : INTEREST SEQUENTIAL ("GMC_2A4#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A4#1")) )
    pay : SEQUENTIAL ("GMC_2A4#1")
-----
    calculate : #GMCWD = BBAL("GMC_2A4#1") - BBAL("2A4#1")
-----
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A4#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A4#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A4#1")) )
    pay : SEQUENTIAL ("GMC_2A4#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A4#1") * COUPON("2A4#1") / 1200 * 24 / 30) )
    pay : AS_INTEREST ("GMC_2A4#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (INTPMT("2A5#1")) )
    pay : INTEREST SEQUENTIAL ("GMC_2A5#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A5#1")) )
    pay : SEQUENTIAL ("GMC_2A5#1")
-----
    calculate : #GMCWD = BBAL("GMC_2A5#1") - BBAL("2A5#1")
-----
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A5#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )

```

```

                                b04f_call.cdi
pay : DECREMENT ( BALANCE "GMC_2A5#1", BY #GMCWD )
-----
!
  when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("GMC_2A5#1")) )
  pay : SEQUENTIAL ("GMC_2A5#1")
-----
  when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A5#1") * COUPON("2A5#1") / 1200 * 24 / 30) )
  pay : AS_INTEREST ("GMC_2A5#1")
-----
  when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (INTPMT("2A6#1")) )
  pay : INTEREST SEQUENTIAL ("GMC_2A6#1")
-----
  when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A6#1")) )
  pay : SEQUENTIAL ("GMC_2A6#1")
-----
calculate : #GMCWD = BBAL("GMC_2A6#1") - BBAL("2A6#1")
-----
ifdef #cmover_3.0f _
  when : IS_TRUE ( CURMONTH LE 59 )
  from : SUBACCOUNT ( #GMCWD )
  pay : WRITEDOWN SEQUENTIAL ("GMC_2A6#1")
-----
!
ifndef #cmover_3.0f _
  when : IS_TRUE ( CURMONTH LE 59 )
  pay : DECREMENT ( BALANCE "GMC_2A6#1", BY #GMCWD )
-----
!
  when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("GMC_2A6#1")) )
  pay : SEQUENTIAL ("GMC_2A6#1")
-----
  when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A6#1") * COUPON("2A6#1") / 1200 * 24 / 30) )
  pay : AS_INTEREST ("GMC_2A6#1")
-----
calculate: #CallBalGrp1 = COLL_BAL(1)
calculate: #CallBalGrp2 = COLL_BAL(2)
calculate: #CallBalGrp3 = COLL_BAL(3)
calculate: #CallBalGrp4 = COLL_BAL(4)
-----
----- SECTION: "OPTR_DEAL"
-----
  from : CASH_ACCOUNT (100)
subject to : CEILING ( #CallBalGrp1 )
  pay : CLASS BALANCE SEQUENTIAL ( "GRP1" )
  pay : CLASS MORE_INTEREST SEQUENTIAL ( "GRP1" )
-----
  from : CLASS ( "GRP1" )
  pay : SEQUENTIAL ( "RESID#1", "1A1#1" )
-----
  from : CLASS ( "GRP1" )
  pay : CLASS BALANCE SEQUENTIAL ( "SUBORD_1" )
-----
subject to : CEILING ( (BBAL("SUBORD_1#1")-BBAL("SUBORD_1")) )
  pay : SEQUENTIAL ( "SUBORD_1#1" )
-----
  from : CLASS ( "SUBORD_1" )

```

```

                                b04f_call.cdi
pay : SEQUENTIAL ( "SUBS#1" )
-----
calculate : #WriteDown1 = BBAL("GRP1")
-----
calculate : #WD_SUBORD_1 = MIN(BBAL("SUBORD_1#1"), #Writedown1)
calculate : #Writedown1 = #Writedown1 - #WD_SUBORD_1
-----
subject to : CEILING ( #WD_SUBORD_1 )
pay : DECREMENT( BALANCE "SUBORD_1#1", BY #WD_SUBORD_1 )
-----
from : SUBACCOUNT ( #Writedown1 )
pay : WRITEDOWN PRO_RATA ( "RESID#1"; "1A1#1" )
-----
from : CASH_ACCOUNT (100)
subject to : CEILING ( #CallBalGrp2)
pay : CLASS BALANCE SEQUENTIAL ( "GRP2" )
pay : CLASS MORE_INTEREST SEQUENTIAL ( "GRP2" )
-----
from : CLASS ( "GRP2" )
pay : SEQUENTIAL ( "2A1#1", "2A2#1", "2A3#1", "2A4#1", "2A5#1", "2A6#1",
"2A7#1" )
-----
from : CLASS ( "GRP2" )
pay : CLASS BALANCE SEQUENTIAL ( "SUBORD_2" )
-----
subject to : CEILING ( (BBAL("SUBORD_2#1")-BBAL("SUBORD_2"))) )
pay : SEQUENTIAL ( "SUBORD_2#1" )
-----
from : CLASS ( "SUBORD_2" )
pay : SEQUENTIAL ( "SUBS#1" )
-----
when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A1#1"))) )
pay : SEQUENTIAL ("GMC_2A1#1")
-----
calculate : #GMCWD = BBAL("GMC_2A1#1") - BBAL("2A1#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A1#1")
    -----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A1#1", BY #GMCWD )
    -----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A1#1"))) )
    pay : SEQUENTIAL ("GMC_2A1#1")
    -----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A1#1") * COUPON("2A1#1") / 1200 * 0 / 30) )
    pay : AS_INTEREST ("GMC_2A1#1")
    -----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A2#1"))) )
    pay : SEQUENTIAL ("GMC_2A2#1")
    -----
    calculate : #GMCWD = BBAL("GMC_2A2#1") - BBAL("2A2#1")
    -----

```

```

                                b04f_call.cdi
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A2#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A2#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A2#1")) )
    pay : SEQUENTIAL ("GMC_2A2#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A2#1") * COUPON("2A2#1") / 1200 * 0 / 30) )
    pay : AS_INTEREST ("GMC_2A2#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A3#1")) )
    pay : SEQUENTIAL ("GMC_2A3#1")
-----
    calculate : #GMCWD = BBAL("GMC_2A3#1") - BBAL("2A3#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A3#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A3#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A3#1")) )
    pay : SEQUENTIAL ("GMC_2A3#1")
-----
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("2A3#1") * COUPON("2A3#1") / 1200 * 24 / 30) )
    pay : AS_INTEREST ("GMC_2A3#1")
-----
    when : IS_TRUE ( CURMONTH LE 59 )
    subject to : CEILING ( (PRINCPMT("2A4#1")) )
    pay : SEQUENTIAL ("GMC_2A4#1")
-----
    calculate : #GMCWD = BBAL("GMC_2A4#1") - BBAL("2A4#1")
-----
ifdef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    from : SUBACCOUNT ( #GMCWD )
    pay : WRITEDOWN SEQUENTIAL ("GMC_2A4#1")
-----
!
ifndef #cmover_3.0f _
    when : IS_TRUE ( CURMONTH LE 59 )
    pay : DECREMENT ( BALANCE "GMC_2A4#1", BY #GMCWD )
-----
!
    when : IS_TRUE ( CURMONTH EQ 59 )
    subject to : CEILING ( (BBAL("GMC_2A4#1")) )

```

```

                                b04f_call.cdi
pay : SEQUENTIAL ("GMC_2A4#1")
-----
when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A4#1") * COUPON("2A4#1") / 1200 * 24 / 30) )
pay : AS_INTEREST ("GMC_2A4#1")
-----
when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A5#1")) )
pay : SEQUENTIAL ("GMC_2A5#1")
-----
calculate : #GMCWD = BBAL("GMC_2A5#1") - BBAL("2A5#1")
-----
ifdef #cmover_3.0f _
when : IS_TRUE ( CURMONTH LE 59 )
from : SUBACCOUNT ( #GMCWD )
pay : WRITEDOWN SEQUENTIAL ("GMC_2A5#1")
-----
!
ifndef #cmover_3.0f _
when : IS_TRUE ( CURMONTH LE 59 )
pay : DECREMENT ( BALANCE "GMC_2A5#1", BY #GMCWD )
-----
!
when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("GMC_2A5#1")) )
pay : SEQUENTIAL ("GMC_2A5#1")
-----
when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A5#1") * COUPON("2A5#1") / 1200 * 24 / 30) )
pay : AS_INTEREST ("GMC_2A5#1")
-----
when : IS_TRUE ( CURMONTH LE 59 )
subject to : CEILING ( (PRINCPMT("2A6#1")) )
pay : SEQUENTIAL ("GMC_2A6#1")
-----
calculate : #GMCWD = BBAL("GMC_2A6#1") - BBAL("2A6#1")
-----
ifdef #cmover_3.0f _
when : IS_TRUE ( CURMONTH LE 59 )
from : SUBACCOUNT ( #GMCWD )
pay : WRITEDOWN SEQUENTIAL ("GMC_2A6#1")
-----
!
ifndef #cmover_3.0f _
when : IS_TRUE ( CURMONTH LE 59 )
pay : DECREMENT ( BALANCE "GMC_2A6#1", BY #GMCWD )
-----
!
when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("GMC_2A6#1")) )
pay : SEQUENTIAL ("GMC_2A6#1")
-----
when : IS_TRUE ( CURMONTH EQ 59 )
subject to : CEILING ( (BBAL("2A6#1") * COUPON("2A6#1") / 1200 * 24 / 30) )
pay : AS_INTEREST ("GMC_2A6#1")
-----
calculate : #WriteDown2 = BBAL("GRP2")
-----
calculate : #WD_SUBORD_2 = MIN(BBAL("SUBORD_2#1"), #writedown2)
calculate : #Writedown2 = #Writedown2 - #WD_SUBORD_2
-----
subject to : CEILING ( #WD_SUBORD_2 )
pay : DECREMENT( BALANCE "SUBORD_2#1", BY #WD_SUBORD_2 )

```

b04f_call.cdi

```
-----
      from : SUBACCOUNT ( #Writedown2 )
      pay : WRITEDOWN PRO_RATA ( "2A1#1"; "2A2#1"; "2A3#1"; "2A4#1"; "2A5#1";
"2A6#1"; "2A7#1" )
-----
```

```
-----
      from : CASH_ACCOUNT (100)
subject to : CEILING ( #CallBalGrp3)
      pay : CLASS BALANCE SEQUENTIAL ( "GRP3" )
      pay : CLASS MORE_INTEREST SEQUENTIAL ( "GRP3" )
-----
```

```
-----
      from : CLASS ( "GRP3" )
      pay : SEQUENTIAL ( "3A1#1" )
-----
```

```
-----
      from : CLASS ( "GRP3" )
      pay : CLASS BALANCE SEQUENTIAL ( "SUBORD_3" )
-----
```

```
subject to : CEILING ( (BBAL("SUBORD_3#1")-BBAL("SUBORD_3"))) )
      pay : SEQUENTIAL ( "SUBORD_3#1" )
-----
```

```
-----
      from : CLASS ( "SUBORD_3" )
      pay : SEQUENTIAL ( "SUBS#1" )
-----
```

```
calculate : #WriteDown3 = BBAL("GRP3")
-----
```

```
calculate : #WD_SUBORD_3 = MIN(BBAL("SUBORD_3#1"), #Writedown3)
calculate : #Writedown3 = #Writedown3 - #WD_SUBORD_3
-----
```

```
subject to : CEILING ( #WD_SUBORD_3 )
      pay : DECREMENT( BALANCE "SUBORD_3#1", BY #WD_SUBORD_3 )
-----
```

```
-----
      from : SUBACCOUNT ( #Writedown3 )
      pay : WRITEDOWN SEQUENTIAL ( "3A1#1" )
-----
```

```
-----
      from : CASH_ACCOUNT (100)
subject to : CEILING ( #CallBalGrp4)
      pay : CLASS BALANCE SEQUENTIAL ( "GRP4" )
      pay : CLASS MORE_INTEREST SEQUENTIAL ( "GRP4" )
-----
```

```
-----
      from : CLASS ( "GRP4" )
      pay : SEQUENTIAL ( "4A1#1" )
-----
```

```
-----
      from : CLASS ( "GRP4" )
      pay : CLASS BALANCE SEQUENTIAL ( "SUBORD_4" )
-----
```

```
subject to : CEILING ( (BBAL("SUBORD_4#1")-BBAL("SUBORD_4"))) )
      pay : SEQUENTIAL ( "SUBORD_4#1" )
-----
```

```
-----
      from : CLASS ( "SUBORD_4" )
      pay : SEQUENTIAL ( "SUBS#1" )
-----
```

```
calculate : #WriteDown4 = BBAL("GRP4")
-----
```

```
calculate : #WD_SUBORD_4 = MIN(BBAL("SUBORD_4#1"), #Writedown4)
calculate : #Writedown4 = #Writedown4 - #WD_SUBORD_4
-----
```

```
subject to : CEILING ( #WD_SUBORD_4 )
      pay : DECREMENT( BALANCE "SUBORD_4#1", BY #WD_SUBORD_4 )
-----
```

```
-----
      from : SUBACCOUNT ( #Writedown4 )
      pay : WRITEDOWN SEQUENTIAL ( "4A1#1" )
-----
```

b04f_call.cdi

```

!
Schedule "SHIFT1%"
Declare
SHIFTINT GROUP 1
84      100%
96      70%
108     60%
120     40%
132     20%
144     0%
!
!
Schedule "SHIFT2%"
Declare
SHIFTINT GROUP 2
84      100%
96      70%
108     60%
120     40%
132     20%
144     0%
!
!
Schedule "SHIFT3%"
Declare
SHIFTINT GROUP 3
84      100%
96      70%
108     60%
120     40%
132     20%
144     0%
!
!
Schedule "SHIFT4%"
Declare
SHIFTINT GROUP 4
84      100%
96      70%
108     60%
120     40%
132     20%
144     0%
!
!
Collateral
!
!      Factor      --Delay--
! Type   Date      P/Y    BV    Use BV for 0
! WL 20040601      9999 9999  FALSE
!
! Pool#  Type      Gross      Current      Original  --Fee--  Maturity Orig  ARM
!      Gross #mos  #mos    P#mos    P#mos    Life    Reset Life    Max    Look
!      Margin ToRst RstPer ToRst  RstPer Cap    Cap    Floor  Negam Back
!! BEGINNING OF COLLATERAL
M      1      "1"      WL      00      WAC      4.5444382129 (      1066826.43 /
1066826.43 );      1066826.43      0.3750000000 0.3750000000
352:8      352:8      360 NO_CHECK ARM LIBOR_1YR      2.2500000000
29 12 SYNC_INT      10.5444382129      2.0000000000      2.2500000000      0      0
ORIG_GROSSRATE 4.5444382129
GROUP 1      TEASER
M      2      "1"      WL      00      WAC      4.5000000000 (      362485.72 /

```

b04f_call.cdi

```

362485.72 ); 362485.72 0.3750000000 0.3750000000
354:6 354:6 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
31 12 SYNC_INT 10.5000000000 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 4.5000000000

GROUP 1 TEASER
M 3 "1" WL 00 WAC 4.7500000000 ( 615866.29 /
615866.29 ); 615866.29 0.3750000000 0.3750000000
356:4 356:4 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
33 12 SYNC_INT 10.7500000000 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 4.7500000000

GROUP 1 TEASER
M 4 "1" WL 00 WAC 3.9665503634 ( 6375274.01 /
6375274.01 ); 6375274.01 0.3750000000 0.3750000000
358:2 358:2 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
35 12 SYNC_INT 9.9665503634 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 3.9665503634

GROUP 1 TEASER
M 5 "1" WL 00 WAC 4.0298692227 ( 165409098.04 /
165409098.04 ); 165409098.04 0.3750000000 0.3750000000
358:1 358:1 359 NO_CHECK ARM LIBOR_1YR 2.2500000000
36 12 SYNC_INT 10.0298692227 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 4.0298692227

GROUP 1 TEASER
M 6 "1" WL 00 WAC 4.4452795773 ( 55537325.37 /
55537325.37 ); 55537325.37 0.3750000000 0.3750000000
360:0 360:0 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
37 12 SYNC_INT 10.4452795773 2.0000000000 2.2500000000 0 0
ORIG_GROSSRATE 4.4452795773

GROUP 1 TEASER
M 7 "2" WL 00 WAC 5.0000000000 ( 596992.32 /
596992.32 ); 596992.32 0.2500000000 0.2500000000
347:13 347:13 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
48 12 SYNC_INT 10.0000000000 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 5.0000000000

GROUP 2 TEASER
M 8 "2" WL 00 WAC 4.5428934028 ( 1097688.53 /
1097688.53 ); 1097688.53 0.2500000000 0.2500000000
351:9 351:9 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
52 12 SYNC_INT 9.5428934028 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.5428934028

GROUP 2 TEASER
M 9 "2" WL 00 WAC 4.1926161867 ( 3471014.97 /
3471014.97 ); 3471014.97 0.2500000000 0.2500000000
309:8 309:8 317 NO_CHECK ARM LIBOR_1YR 2.2500000000
53 12 SYNC_INT 9.1926161867 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.1926161867

GROUP 2 TEASER
M 10 "2" WL 00 WAC 4.8560092813 ( 822898.16 /
822898.16 ); 822898.16 0.2500000000 0.2500000000
354:6 354:6 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
55 12 SYNC_INT 9.8560092813 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.8560092813

GROUP 2 TEASER
M 11 "2" WL 00 WAC 4.5531677162 ( 4037666.05 /
4037666.05 ); 4037666.05 0.2500000000 0.2500000000
355:5 355:5 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
56 12 SYNC_INT 9.5531677162 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.5531677162

GROUP 2 TEASER
M 12 "2" WL 00 WAC 4.8750000000 ( 337322.48 /
337322.48 ); 337322.48 0.2500000000 0.2500000000
356:4 356:4 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
57 12 SYNC_INT 9.8750000000 2.0000000000 2.2500000000 0 0

```

b04f_call.cdi

```

INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.8750000000
GROUP 2 TEASER
M 13 "2" WL 00 WAC 4.8003425711 ( 1113206.23 /
1113206.23 ); 1113206.23 0.2500000000 0.2500000000
357:3 357:3 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
58 12 SYNC_INT 9.8003425711 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.8003425711
GROUP 2 TEASER
M 14 "2" WL 00 WAC 4.3982787548 ( 24277616.88 /
24277616.88 ); 24277616.88 0.2500000000 0.2500000000
353:2 353:2 355 NO_CHECK ARM LIBOR_1YR 2.2500000000
59 12 SYNC_INT 9.3982787548 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.3982787548
GROUP 2 TEASER
M 15 "2" WL 00 WAC 4.3685480805 ( 348613931.93 /
348613931.93 ); 348613931.93 0.2500000000 0.2500000000
353:1 353:1 354 NO_CHECK ARM LIBOR_1YR 2.2500000000
60 12 SYNC_INT 9.3685480805 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.3685480805
GROUP 2 TEASER
M 16 "2" WL 00 WAC 4.6153520857 ( 84169387.12 /
84169387.12 ); 84169387.12 0.2500000000 0.2500000000
356:0 356:0 356 NO_CHECK ARM LIBOR_1YR 2.2500000000
61 12 SYNC_INT 9.6153520857 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.6153520857
GROUP 2 TEASER
M 17 "2-IO" WL 00 WAC 4.6250000000 ( 390400.00 /
390400.00 ); 390400.00 0.2500000000 0.2500000000
355:5 355:5 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
56 12 SYNC_INT 9.6250000000 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.6250000000 AMORT NONE FOR 60
GROUP 2 TEASER
M 18 "2-IO" WL 00 WAC 5.3750000000 ( 333701.00 /
333701.00 ); 333701.00 0.2500000000 0.2500000000
356:4 356:4 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
57 12 SYNC_INT 10.3750000000 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 5.3750000000 AMORT NONE FOR 60
GROUP 2 TEASER
M 19 "2-IO" WL 00 WAC 4.2342266461 ( 3091257.60 /
3091257.60 ); 3091257.60 0.2500000000 0.2500000000
358:2 358:2 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
59 12 SYNC_INT 9.2342266461 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.2342266461 AMORT NONE FOR 60
GROUP 2 TEASER
M 20 "2-IO" WL 00 WAC 4.3970325002 ( 450419045.56 /
450419045.56 ); 450419045.56 0.2500000000 0.2500000000
359:1 359:1 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
60 12 SYNC_INT 9.3970325002 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.3970325002 AMORT NONE FOR 60
GROUP 2 TEASER
M 21 "2-IO" WL 00 WAC 4.6341836373 ( 168511754.13 /
168511754.13 ); 168511754.13 0.2500000000 0.2500000000
360:0 360:0 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
61 12 SYNC_INT 9.6341836373 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.6341836373 AMORT NONE FOR 60
GROUP 2 TEASER
M 22 "3" WL 00 WAC 5.7500000000 ( 135661.59 /
135661.59 ); 135661.59 0.2500000000 0.2500000000
167:13 167:13 180 NO_CHECK ARM LIBOR_1YR 2.2500000000
48 12 SYNC_INT 10.7500000000 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 5.7500000000
GROUP 3 TEASER
M 23 "3" WL 00 WAC 5.1250000000 ( 212163.31 /

```

b04f_call.cdi

```

212163.31 ); 212163.31 0.2500000000 0.2500000000
349:11 349:11 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
50 12 SYNC_INT 10.1250000000 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 5.1250000000
GROUP 3 TEASER
M 24 "3" WL 00 WAC 4.7500000000 ( 268887.51 /
268887.51 ); 268887.51 0.2500000000 0.2500000000
352:8 352:8 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
53 12 SYNC_INT 9.7500000000 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.7500000000
GROUP 3 TEASER
M 25 "3" WL 00 WAC 5.5000000000 ( 320193.22 /
320193.22 ); 320193.22 0.2500000000 0.2500000000
353:7 353:7 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
54 12 SYNC_INT 10.5000000000 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 5.5000000000
GROUP 3 TEASER
M 26 "3" WL 00 WAC 4.5000000000 ( 147805.10 /
147805.10 ); 147805.10 0.2500000000 0.2500000000
359:1 359:1 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
60 12 SYNC_INT 9.5000000000 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.5000000000
GROUP 3 TEASER
M 27 "3-IO" WL 00 WAC 4.8750000000 ( 226000.00 /
226000.00 ); 226000.00 0.2500000000 0.2500000000
354:6 354:6 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
55 12 SYNC_INT 9.8750000000 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.8750000000 AMORT NONE FOR 60
GROUP 3 TEASER
M 28 "3-IO" WL 00 WAC 4.4582975399 ( 3582237.53 /
3582237.53 ); 3582237.53 0.2500000000 0.2500000000
357:3 357:3 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
58 12 SYNC_INT 9.4582975399 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.4582975399 AMORT NONE FOR 60
GROUP 3 TEASER
M 29 "3-IO" WL 00 WAC 4.3663695038 ( 9233678.39 /
9233678.39 ); 9233678.39 0.2500000000 0.2500000000
358:2 358:2 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
59 12 SYNC_INT 9.3663695038 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.3663695038 AMORT NONE FOR 60
GROUP 3 TEASER
M 30 "3-IO" WL 00 WAC 4.4423488603 ( 150127639.25 /
150127639.25 ); 150127639.25 0.2500000000 0.2500000000
359:1 359:1 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
60 12 SYNC_INT 9.4423488603 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.4423488603 AMORT NONE FOR 60
GROUP 3 TEASER
M 31 "3-IO" WL 00 WAC 4.7355598691 ( 51027081.71 /
51027081.71 ); 51027081.71 0.2500000000 0.2500000000
360:0 360:0 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
61 12 SYNC_INT 9.7355598691 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.7355598691 AMORT NONE FOR 60
GROUP 3 TEASER
M 32 "4" WL 00 WAC 4.9017954660 ( 5971807.97 /
5971807.97 ); 5971807.97 0.2500000000 0.2500000000
358:2 358:2 360 NO_CHECK ARM LIBOR_1YR 2.2500000000
83 12 SYNC_INT 9.9017954660 2.0000000000 2.2500000000 0 0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.9017954660
GROUP 4 TEASER
M 33 "4" WL 00 WAC 4.8481000167 ( 84798305.31 /
84798305.31 ); 84798305.31 0.2500000000 0.2500000000
355:1 355:1 356 NO_CHECK ARM LIBOR_1YR 2.2500000000
84 12 SYNC_INT 9.8481000167 2.0000000000 2.2500000000 0 0

```

```

                                b04f_call.cdi
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 4.8481000167
GROUP 4      TEASER
M           34      "4"      WL      00      WAC      5.1939228940 (      18286720.91 /
18286720.91 );      18286720.91      0.2500000000 0.2500000000
           360:0      360:0      360 NO_CHECK ARM LIBOR_1YR      2.2500000000
85  12 SYNC_INT      10.1939228940      2.0000000000      2.2500000000      0      0
INIT_PERCAP 5.0000000000 ORIG_GROSSRATE 5.1939228940
GROUP 4      TEASER

```